RIVISTA UFFICIALMENTE RICONOSCIUTA DA COMMODORE ITALIANA

SPECIALE CAMPIONATORI AUDIO

IN COLLABORAZIONE CON AMIGA

ANNO 4 - N. 23
MAGGIO 1991
L. 14.000
Frs. 21.00

AMIGA MAGAZINE

IL MENSILE JACKSON PER GLI UTENTI DI AMIGA

- AMIGA IN TELEVISIONE
- IMAGINE
- ARRIVA LA REALTA' VIRTUALE
- SCHEDA MULTI-PORT A2232
- HAND SCANNER JS-105-1M
- AMIGA COSTRUISCE UN MONDO
- TOTOMANIA
- KCS 3.0: IL PROFESSIONISTA DEL MIDI
- DISK DRIVE MASTER 3D-1A
- AMIGA E L'INTELLIGENZA ARTIFICIALE
- ON DISK: 10 FANTASTICI PROGRAMMI

INSERTO TransAction
LE PAGINE DEL PROGRAMMATORE

© PRODUZIONE EQUART 1990

SPEDIZIONE IN ABBONAMENTO POSTALE GRUPPO III/70 TAXE PERCUE (TASSA RISCOSSA) MILANO CMP ROSERIO

RADIO CAPITAL

SINTONIZZATI SUGLI ANNI '70

Direttore Responsabile: Paolo Reina
Coordinamento Tecnico e Redazionale: Massimiliano Anticoli
Tel 02 / 6948260
Redazione: Romano Tenca (TransAction) - Lucio Bragagnolo
Segreteria di redazione e coordinamento estero: Elena Ferré
Tel 02/6948254
Art Director: Marcello Longhini
Copertina, Grafica: Cristina Turra
Impaginazione elettronica: DTP Studio - Alessandro Fiore
Collaboratori: Mirco Baiardi, Gianni Biagini, Daniele Cassanelli, Alberto Geneletti, Aldo e Andrea Laus, Diego Montefusco, Stefano Paganini, Cesare Palmieri, Gabriele Ponte, Paul Rigby, Stefano Riva (On Disk), Nicola Salmoria, Marco Tortolina, Sebastiano Vigna, Andrew Walrond, Marco Zandonadi
Corrispondente dagli U.S.A.: Marshal M Rosenthal
British Correspondent: Derek Dela Fuente

Group Publisher: Pierantonio Palerma
Publisher Area Consumer: Filippo Canavese
Coordinamento Operativo: Sarah Platero
Pubblicità: Ambrogio Isacchi - Tel. 02 / 6948218
Direzione Sviluppo Pubblicità: Walter Bussolera
Direzione Marketing e Promotion: Filippo Canavese

SEDE LEGALE
Via P Mascagni, 14 - 20122 Milano

DIREZIONE - REDAZIONE
Via Pola, 9 - 20124 Milano - Tel 02/69481
Fax 02/6948238 Telex 316213 REINA I

PUBBLICITA'
Via Pola, 9 - 20124 Milano - Tel 02/6948254
ROMA - LAZIO E CENTRO SUD
Via Lago di Tana, 16 - 00199 Roma
Tel 06/8380547 - Fax 06/8380637
EMILIA ROMAGNA
Giuseppe Pintor - Via della Chiesa, 1 - 40060 Toscanella (BO)
Tel. 051/387790 - Fax. 051/310875
TOSCANA
Camilla Parenti - Publindustria - Via S. Antonio, 22 - 56125 Pisa
Tel 050/47441 - Fax 050/49451

INTERNATIONAL MARKETING
Tel 02/6948233

DIREZIONE AMMINISTRATIVA
Via Rosellini, 12 - 20124 Milano Tel 02/69481
Fax 02/6948238

UFFICIO ABBONAMENTI
Via Amendola, 45-20037 Paderno Dugnano (MI) - Fax 02/99042386
Telex 333436 GEJ IT - Tel.:02/99043119-127-133 (nei giorni di martedì, mercoledì, giovedì 14 30 - 17 30)

Prezzo della rivista L 14 000 prezzo arretrato L 28 000
Non saranno evase richieste di numeri arretrati antecedenti due anni dal numero in corso
Abbonamento annuo Italia L 123 200, Estero L 246 400
I versamenti vanno indirizzati a
Gruppo Editoriale Jackson SpA
Via Rosellini, 12 - 20124 Milano, mediante l'emissione di assegno bancario o per contanti L'abbonamento può essere sottoscritto anche utilizzando il c/c postale 11666203

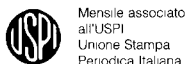
CONSOCIATE ESTERE
GEJ Publishing Group Inc Los Altos Hills
27910 Roble Blanco
94022 California - Tel (001-415-9492028)
Grupo Editorial Jackson - Conde de Penalver, 52
28006 Madrid - Tel 0034/14017365

Stampa: F B M (Gorgonzola)
Fotolito: Foligraph (Milano)
Distribuzione: Sodip - Via Zuretti, 25 - 20125 Milano

Il Gruppo Editoriale Jackson è iscritto al Registro Nazionale della stampa al N 117 Vol 2 foglio 129 in data 17/8/1982.
Spedizione in abbonamento postale gruppo III/70
Aut Trib' di Milano n 102 del 22/2/1988

Parte degli articoli sono tradotti da **Compute 1990/91** su autorizzazione di **Compute Publications International, Ltd.**

Mensile associato all'USPI Unione Stampa Periodica Italiana

Consorzio Stampa Specializzata Tecnica

Testata aderente al C S S.T non soggetta a certificazione obbligatoria per la presenza pubblicitaria inferiore al 10%

Editoriale

Teoria sui massimi sistemi

Ultimamente mi è capitato di partecipare a una di quelle simpaticissime cene, pizza & birra, con alcuni collaboratori della nostra rivista.

La discussione, molto interessante, verteva sulla potenza dei computer in circolazione.

Il risultato ? Naturale, forever Amiga ma non perché siamo innamorati di questo stupendo computer, ma per le ragioni che ora vi spiegherò.

Macintosh: interfaccia utente molto meno user-friendly di Amiga, programmi molto ma molto cari, poco creativo, poco programmabile, pochi giochi e prezzo del modello entry-level decisamente molto elevato.

PC Compatibile: non tutti compatibili al 100%, musica scarsissima, programmi decisamente non entusiasmanti, macchina aperta per la programmazione ma limitata, giochi pochi e molto brutti, prezzi dei programmi alti, prezzi dei modelli entry-level veloci decisamente troppo costosi, anche se questa tendenza si sta modificando molto rapidamente.

Infine, Amiga. la nostra stupenda macchina, ad un prezzo decisamente basso, offre potenzialità elevate. Per alcuni lavori di produttività personale, è la macchina ideale: prezzi dei programmi non molto alti, hardware aggiuntivo disponibile e, sempre il dio-denaro, a poco prezzo, molti giochi e quasi tutti stupendi, programmazione fantastica e...

Be', credo che molti di Voi concorderranno con la mia opinione !!

A questo punto non mi resta che darvi appuntamento al prossimo mese!

Arrivederci in edicola.

Massimiliano Anticoli

ANNO 4 - NUMERO 23
MAGGIO 1991

Sommario

Contatto...

Ciao a tutti,
sono un felicissimo possessore di un A500 con A590, A501, stampante Epson, un drive esterno, 2 Mb di Fast RAM e ovviamente, dato che sono sysop del Bit Exchange Box Club BBS, un modem a 2400 bps. Sono d'obbligo i complimenti alla rivista Amiga Magazine e al suo ottimo inserto TransAction che mi ha aiutato molto a cancellare dalla mente molti dubbi sul linguaggio assembler (BENE!, BRAVI!). Sono un po' meno contento del disco allegato...

Ho il mio Amiga da qualche anno e negli ultimi tempi ho riscontrato alcuni problemi, ve ne elenco alcuni:

1) L'espansione di memoria A501 (100% originale Commodore) mi fa andare in Guru Amiga molto frequentemente sempre con lo stesso codice d'errore (00000004 = illegal opcode). Ho pensato che la scheda non faccia bene contatto, perché se picchio sulla scrivania o direttamente su Amiga (anche molto delicatamente con il solo indice) questo mi va in crash. Alcune volte lo accendo e al posto di andare in boot appare una schermata gialla con il led che lampeggia (si lo so, su Transactor è apparso una articolo che parlava di questi colori...). Tutto questo NON accade se tolgo l'espansione di memoria A501. Vorrei sapere da voi se la mia teoria può essere giusta o se è un BUG di Amiga.

2) L'orologio dell'A501 e quello di Amiga danno i numeri... Anche se tolgo A501 la data rimane su <unset><unset><unset>, alcune volte si inventa la data. (???)

3) Se clicco su un'icona che si trova nella parte inferiore dello schermo (quella che offre il PAL in più del NTSC) il puntatore del mouse di Amiga rimane fisso sullo schermo (dovrebbe scomparire) e appare (mobile) quello con le due ZZ (= sto caricando).

Ho portato a far controllare Amiga, ma mi hanno riferito che è tutto normale...

Comunque, sia il problema del clock che non tiene la data (o se la inventa) e il problema del mouse non mi danno molto fastidio, invece quello di diver lasciare 1/2 Mb di RAM fuori da Amiga non mi rallegra molto...

Se potete in qualche modo aiutarmi io ne sarò ben felice !

Christian Bianchi

Caro Christian,
ti ringraziamo vivamente sia per i complimenti alla rivista che per le critiche e passiamo subito ai tuoi problemi.

Del tuo primo problema è da ritenersi responsabile l'espansione di memoria A501: probabilmente si tratta di un contatto indesiderato dei pin di collegamento tra l'espansione e Amiga (la schermata iniziale colorata conferma questa ipotesi).

Anche il secondo problema che ci poni è da imputarsi all'A501 e in particolare alla batteria tampone (da sostituire con una nuova).

Il terzo punto della tua lettera ci è ormai divenuto familiare: è il problema del Fat Agnus che in presenza di un'espansione di memoria da questi problemi.

Comunque ti consigliamo caldamente di portare il tuo A500 in un centro di Assitenza Commodore e di far verificare bene l'A501.

Sempre assistenza

Spett. readzione, sono un vostro abbonato e possego un Amiga 2000 da poco più di un anno. Per mia sfortuna, a causa di un dischetto deformato, uno dei due drive del mio Amiga non funziona più regolarmente, lo stesso segnala spesso, su alcuni dischetti, degli errori di lettura e/o scrittura inesistenti. Leggendo un vostro articolo sul numero 11 di Amiga Magazine nella rubrica "Il Tecnico Risponde", dedicato al drive di Amiga, ho pensato (male!) di recarmi presso la ditta Computer Lab per sottoporre il mio problema. Al fine di consentire ai tecnici della stessa un test immediato del mio drive, l'ho smontato e mi sono presentato fiducioso alla Computer Lab. A ricevermi c'era una signoria dal fare un po' scostante, ho spiegato alla stessa il problema e rimasi perplesso dal suo atteggiamento nei confronti di quell'oggetto, sembrava che stesse osservando per la prima volta in vita sua una pietra di Marte! Come un bimbo di fronte ad un giocattolo misterioso, incominciò a toccarlo con curiosità raddrizzò il led rosso di 90 gradi, con il rischio di troncare in due i sottili fili che lo legano al telaio. Perplesso dalla mia richiesta, si diresse, con il il drive in mano, verso un tecnico in fondo al laboratorio, questi gli sussurrò qualcosa e ritornò a me con fare ancor più scostante. Mi disse che il laboratorio generalmente non esegue queste riparazioni. Spiegai che il danno poteva essere minimo in quanto il drive a volte leggeva e a volte no, forse poteva essere un problema di allineamento di testine. Mi rispose che il laboratorio non opera sul drive con problemi di allineamento! A questo punto chiesi se dovevo buttarlo dalla finestra o si fossero degnati, quanto meno, di trattenerlo per un esame. Con tono di sufficienza mi disse: "se vuole... " e aggiunse "comunque non si porta a riparare un componente smontato e in queste condizioni". Questa fu la goccia che fece traboccare il vaso, sappiamo tutti che il povero utente Commodore è costretto più volte ad ingegnarsi

tecnico, qualsiasi periferica viene venduta con uno scarno libretto di istruzione per il montaggio e un cordiale "arrangiati" (drive compreso). Recentemente ho acquistato una scheda della GVP A3001 con 68030 Mb di RAM e un hard disk Quantum, come sempre ho dovuto arrangiarmi da solo per il montaggio e l'installazione del tutto, con il rischio di compromettere una spesa di qualche milione di lire. Fortunatamente tutto funziona egregiamente. Facendo le dovute rimostranze mi ripresi il mio drive e decisi che non avrei mai più messo piede in quel posto. Mi domando perché il cliente Commodore deve sempre essere trattato in questo modo? E' mai possibile che Amiga debba servire solo a qualche ragazzino desideroso di cimentarsi in giochi? Sappiamo bene che la Commodore ha passato guai seri a suo tempo, mi domando comunque come, con una politica di mercato così raffazzonata, non sia ancora fallita! Da tempo mi accarezza l'idea di acquistare un Macintosh e godere dei servizi d'assistenza professionale allo stesso collegati, quello che mi ha sempre frenato è il prezzo, visto comunque quello che ho speso su Amiga e servizi di "pulcinella" dedicarti all'utente Commodore penso proprio che l'idea divenga presto realtà.

Arduino Puglielli

Ci dispiace per quanto le è accaduto, tuttavia ci risulta che Computer Lab è uno fra i più seri centri d'assistenza Commodore italiani. Probabilmente, c'è stato qualche momento di incomprensione (a dir la verità era meglio che portava tutto e non solo il drive smontato, in maniera tale da collaudare il computer in condizioni di partenza). Non trattenere il drive in questione è stato un gesto di assoluta onestà, poiché il prezzo per "testare" il drive e allineare le testine è pari a un disk drive nuovo. Comunque, nel caso dei manuali ha assolutamente ragione e come vede noi in ogni articolo prendiamo in esame il manuale allegato.

a cura di M. Anticoli e Sheldon Leemon

TDK ITALIA distribuisce i floppy disk

La distribuzione dei floppy disk TDK è recentemente passata alla filiale italiana della nota azienda giapponese.
TDK ITALIA SPA curerà la commercializzazione della gamma di floppy sia attraverso la già consolidata rete di vendita operante nel settore delle cassette audio e video, che attraverso alcuni distributori regionali.
La gamma dei floppy disk TDK comprende attualmente le versioni 5 1/4" e 3 1/2" doppia e alta densità ma verrà presto ampliata con l'inserimento di nuovi prodotti (Data cartridge 8 e 4 mm, floppy colorati, floppy preformattati ecc.). ▲

Accordo Commodore e Pioneer

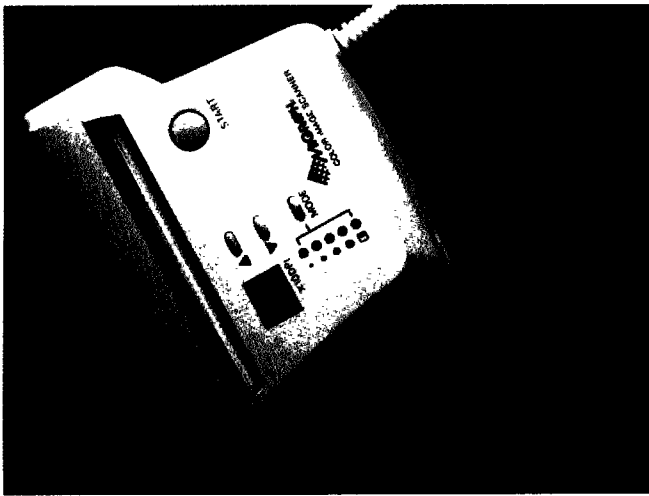
Un accordo tra Commodore Italiana Spa e Pioneer Electronics Italia Spa, azienda giapponese leader nell'elettronica di consumo, da poco entrata anche in Italia nel campo dell'informatica, apre nuove frontiere agli sviluppatori Amiga specializzati in applicazioni multimediali.

modelli), si interfacciano mediante porta seriale ad Amiga, e sono compatibili con AmigaVision per lo sviluppo di applicazioni multimediali. LD 4100 costituisce il prodotto destinato al mercato professionale ad un prezzo estremamente interessante, mentre LD-V8000 è il modello industriale a prestazioni superiori, disponibile attualmente solo in standard NTSC. Gli operatori di settore potranno contattare direttamente: ▲

Commodore Italiana Spa, Viale Fulvio Testi, 280
20126 milano - tel.02-661231

Pioneer Electronics Italia Spa, Via Fantoli, 17
20138 milano - Tel.02-50741

Scanner a colori

La Migraph ha deciso di spazzare via gli scanner in bianco e nero mediante il suo scanner a colori. Facile da usare, bello da vedere, dall'aspetto high-tech e dotato di sufficienti controlli da far sentire ognuno a proprio agio, lo scanner comprende anche software grafico e un'interfaccia grafica semplice da usare. La sua ampia finestra (larga più di 10 cm) può elaborare una cartolina da 15 cm in una sola passata e analizzarla in meno di 12 secondi. Le sue potenti caratteristiche (che ricava dal computer) gli consentono di competere con gli scanner a piano fisso. La risoluzione può variare, a incrementi di 10 dpi, fra il grossolano e l'alta definizione (la si intenda come 400 dpi!). La gamma dei colori è completa, grazie a una risoluzione di 12 bit per pixel: cioè 4096 colori. La fonte luminosa interna permette di analizzare anche in bianco e nero e con scale di grigi e l'output può essere salvato in molti formati appartenenti a diversi computer (come TIFF e PCX).
Uno dei problemi degli scanner manuali è quello di mantenere una mano salda e paziente, per far scorrere l'unità sulla pagina. E' bene sapere che questa necessità è stata sottratta alla mano dell'utente e posta nel territorio della CPU, dove dovrebbe essere. Tutto ciò che si deve fare è attivare il pulsante sullo scanner, mentre si preme dolcemente verso il basso l'unità con un paio di dita: lo scanner produrrà automaticamente un'immagine integra, essendo stato progettato per "rotolare" verso destra, qualsiasi sia la pressione applicata. Per capire veramente cosa significhi, basta confrontare uno degli altri scanner manuali a quello della Migraph. Anche voi vi convertirete immediatamente.
Il software per adattare e convertire le immagini

digitalizzate è abbondante, ma la Migraph accompagna lo scanner con software per il desktop e l'elaborazione delle immagini nei propri pacchetti per IBM, i primi a essere pubblicati. Il pacchetto Amiga (lo stesso scanner, ma con un'interfaccia differente) apparirà a ruota e, secondo un "porta-voce" della Migraph, "comprenderà molto probabilmente un set similare di programmi compatibili con Amiga". Non è invece ancora previsto il supporto per il CDTV. ▲

M. M. Rosenthal

Sicof '91

Una finestra sulle più avanzate tecnologie di riproduzione fotografica.

Si è tenuto a Milano, dal 28 Febbraio al 4 marzo, il Sicof, ovvero l'appuntamento biennale di maggior rilevanza con la fotografia in generale e, in particolare, con le più avanzate tecnologie di riproduzione su carta, ma anche con attrezzature legate al mondo degli audiovisivi.

E' proprio in quest'ottica che compare sulle pagine di questa rivista una notizia legata ad un settore apparentemente così differente ma, in realtà, ora più che mai in collisione con l'informatica. Infatti, i PC e Amiga in particolare, per soddisfare le esigenze di output di settori specifici quali DTP, image processing e rendering da modelli CAD, hanno sempre più fatto uso di tecniche e materiali fotografici ad alto livello. Dall'altro lato, la fotografia ha potuto usufruire di sistemi di editing di immagini digitalizzate e restituite al supporto cartaceo con la massima qualità. La 14esima edizione del Sicof ha visto, al di là della consueta messe di sale di posa popolate da indescrivibili fanciulle, una seppur timida tendenza verso l'immagine digitale: dalla fotocamera alla visualizzazione. Ne sono un esempio la nuova Canon ION RC260, evoluzione della RC 251 già apparsa su Amiga Magazine, e il sistema annunciato dalla Kodak, presente con uno stand stile Caesar's Palace, denominato Photo CD. Tale sistema dovrebbe costituire l'inizio per la conversione verso una completa gestione digitale delle immagini anche a livello di utente domestico.

Il sistema, in sintesi, prevede la digitalizzazione delle immagini fornite dall'utente e la memorizzazione su CD Video, in modo da creare veri e propri CD Album, considerando anche l'eventualità di intervenire qualitativamente sulle immagini stesse: il nome Kodak garantisce. Per la visualizzazione, la Kodak, prevede il lancio di un lettore ad un prezzo estremamente interessate per il mercato home. Oltre a prevedere l'input da immagini su pellicola, il Photo CD, nella versione prototipo, è basato su un sensore CCD da 1.300.000 pixel applicato ad un dorso di una reflex 35 mm, il CCD a sua volta trasferisce le immagini ad una unità hard disk da 200 Mb.

Il sistema è evidentemente professionale ma lascia intravedere futuri sviluppi anche per l'utente dilettante, soprattutto senza scalfire la cultura della pellicola comune sulla quale la stessa Kodak poggia le proprie fondamenta. ▲

Stefano Paganini

Troppi colori?

I colori a 24 bit sono di gran moda. L'Impulse ha rilasciato il suo frame buffer a 24 bit, il Firecracker, sebbene oggi solo 3 o 4 pacchetti software siano in grado di supportarlo (non c'è supporto da parte del Workbench per le schede a colori in alta risoluzione, ne ci sarà ancora per un bel po').

La Progressive Peripherals ha annunciato recentemente la sua scheda Video Blender a 24 bit, si dice che la GVP stia lavorando a un frame buffer a 24 bit e la M.A.S.T. ha presentato il suo economico Colorburst.

Ironicamente, il mondo IBM sembra stia ripensando al suo atteggiamento verso le schede a 24 bit. Queste schede richiedono troppa memoria video ed è difficile manipolare tutti quei bit velocemente, il che significa che le schede sono sia costose che lente.

Alcuni produttori di PC concordano nel ritenere che i colori a 16 bit possono dare dei risultati professionali a un quarto del costo. Persino l'IBM ha recentemente annunciato un nuovo adattatore XGA (ancora un altro standard video? Ebbene sì!) che fornisce 32000 colori sullo schermo, invece dei più ambiziosi 16.7 milioni. ▲

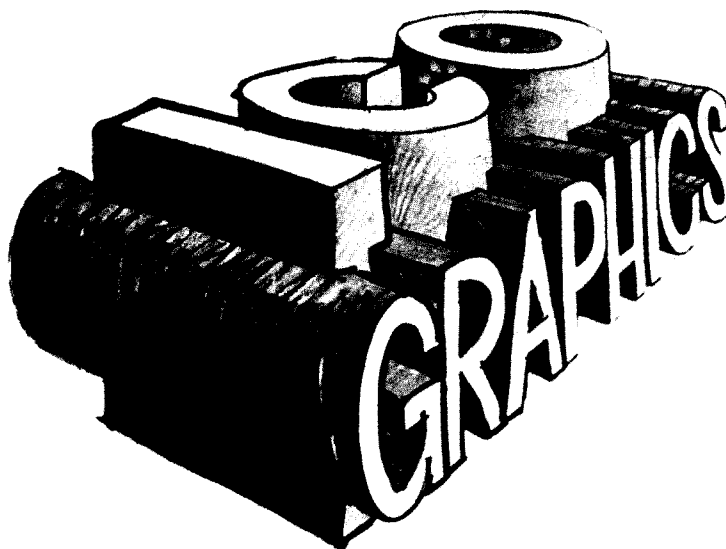
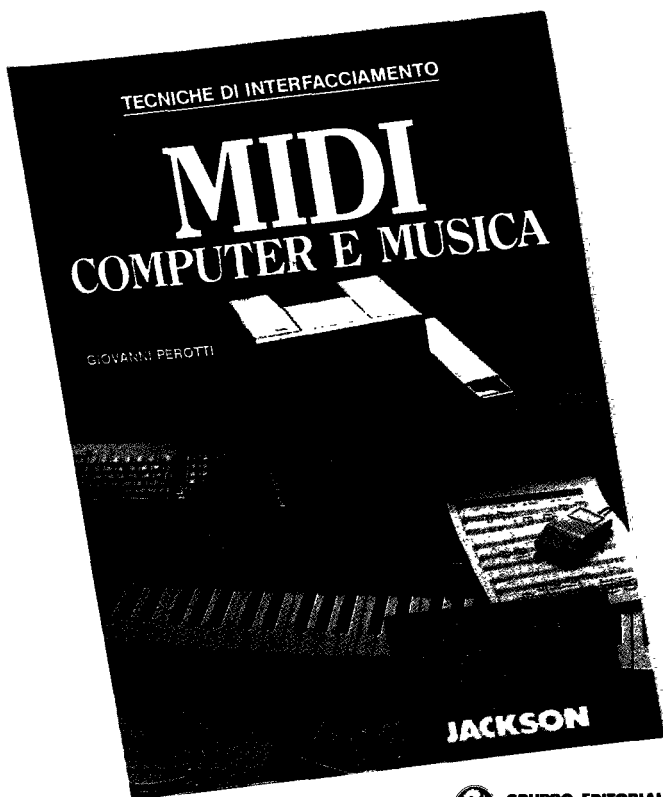
Sheldon Leemon

I.CO. Graphics '91

I.CO. Graphics: uno degli appuntamenti più importanti per il mercato italiano della computer grafica.

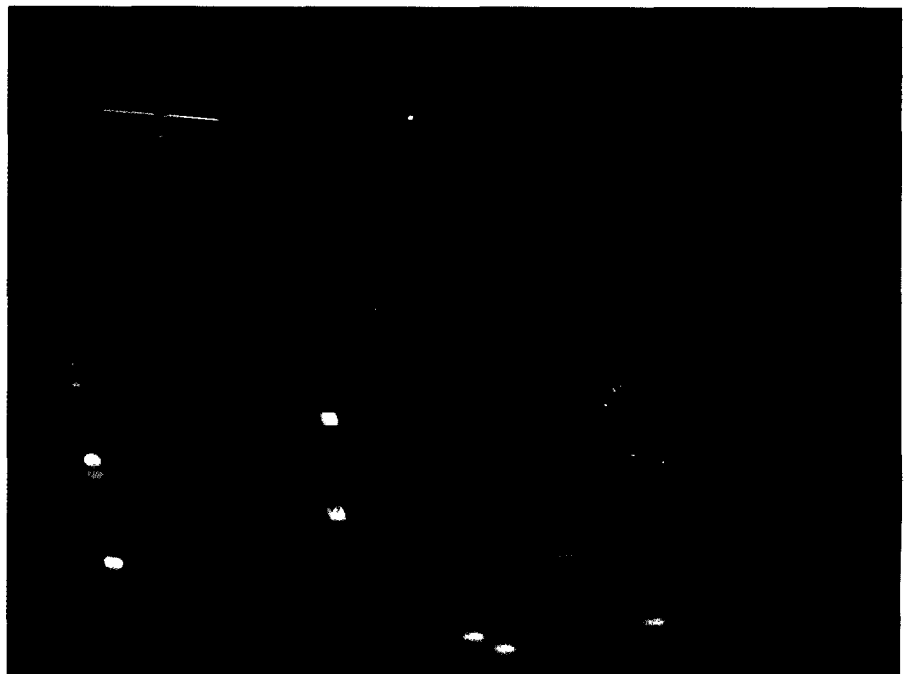
Stefano Paganini

L'edizione di quest'anno, che occupava l'intero padiglione 14 della Fiera su tre piani, oltre alla prevista notevole affluenza di operatori del settore e non, presentava molte novità e conferme di tendenze di mercato già affermate. Grande risalto ha avuto il CAD e in particolare il rendering, per il quale sono disponibili ormai parecchi software e hardware, quali lo standard Renderman sviluppato da Pixar e disponibile ormai su quasi tutte le piattaforme grafiche: tra le ultime MacRenderman e il corrispondente della Autodesk per PC. La prevalenza dei prodotti riguardava PC, Mac e le Workstation Unix. Tra le novità più interessanti ad alto livello, la Silicon Graphics, uno dei nomi leader del settore, esibiva la nuova scheda per PC IRISVision che offre su una singola scheda 16 milioni di colori e una capacità di elaborazione, simile a quella di una stazione IRIS, di 20 Mflops, 90.000 vettori 3D al secondo il tutto tramite il Geometry Engine della stessa Silicon Graphics. In altre parole si tratta di una scheda in grado di realizzare animazioni e rendering in 3D praticamente in tempo reale, con accesso alle librerie software già sviluppate per le altre workstation SGI. In mezzo a tanti PC, tuttavia, Amiga ha occupato un posto di riguardo. Infatti, in luogo dei consueti punti di riferimento per le informazioni, sono stati utilizzati dieci INFO POINT forniti dalla Commodore. Si tratta di postazioni complete formate da un 3000 e dal monitor TouchScreen capacitativo originale Commodore, il tutto già presentato allo scorso SMAU e in questa occasione proficuamente applicato. Ciascun INFO POINT, tramite una applicazione sviluppata sotto AmigaVision, permetteva di avere notizie riguardo al salone, all'ubicazione dei singoli espositori, con ricerca intelligente del nominativo, e, infine, la possibilità di tracciare un percorso ideale dall'INFO POINT allo stand di inte-

resse, realizzando così una "navigazione dinamica", per usare la terminologia ufficiale. Tra le società espositrici segnaliamo la NEC della quale è nota la completa gamma di monitor ad alta risoluzione, non solo per PC ma anche per stazioni Amiga 3000, quali, ad esempio, il best seller 3D o i più "grandi" 4D e 5D, quest'ultimo dedicato al CAD/CAM ad alto livello. Sono state, inoltre, presentate le nuove stampanti, P20 e P30, entrambe a 24 aghi, con prestazioni e prezzi di estremo interesse: 216 cps in draft, 8 font residenti, pratica gestione della carta e alta risoluzione indirizzabile (360x360 dpi). La P20, 80 colonne, costa 870.000+IVA, la P30, 136 colonne con densità di 10 cpi, costa 1.180.000.+IVA. Piccolo ma gremito, lo stand della Polaroid, tra i pionieri dei sistemi di output su pellicola e su slide per PC, che oltre alla consueta versione della PALETTE, esibiva il modello CI3000 Digital Palette, sviluppato esclusivamente per PS/2, con risoluzione massima di 2048x1866 su filmpack 669, comunque torneremo presto su questo argomento per quanto riguarda esclusivamente Amiga. ▲

Amiga in Televisione

Stefan Roda

I più attenti di voi si saranno sicuramente accorti, negli ultimi tempi, della inconfondibile presenza di grafica Amiga in molte trasmissioni televisive, sia nei Network Privati che in Rai. Utilizzata nelle trasmissioni di giochi, redazionali e persino spot pubblicitari, la macchina della Commodore è diventata uno dei mezzi preferiti da tecnici e registi televisivi, per titolazioni, inserti grafici ed effetti speciali tramite genlock. Inoltre, molti studi di produzione privati, utilizzano Amiga per i lavori di titolazione e presentazione di filmati pubblicitari che, ultimamente, molte aziende, dei più disparati settori, inviano alla propria clientela, al posto del classico depliant, sottoforma di videocassetta VHS. Questo successo è principalmente dovuto, a parte le ottime qualità grafiche della macchina, alla flessibilità, praticità e disponibilità di potenti pacchetti, a basso costo, per le elaborazioni grafiche. E in ultima analisi, cosa non certo trascurabile, per l'economicità di un DeskTop Video basato su Amiga. Certamente, un computer dedicato, come, ad esempio, il Symbolics, permette di fare cose incredibili con animazioni 3D come quelle che siamo abituati a vedere in trasmissioni di rilievo ma, è anche vero, che il costo di un simile sistema (hardware/software) può superare il centinaio di milioni (più il costo di realizzazione di ciascuna sigla). Attualmente, purtroppo, non esiste una nuova "vera" scheda grafica a 24 bit (16 milioni di colori) standard per Amiga, quelle attualmente disponibili (Targa, Vista e company) funzionano su Brigeboard AT (installata su Amiga 2000 o 3000) con elevata qualità dell'output grafico ma con il principale difetto di una certa lentezza (rispetto ai sistemi professionali) e soprattutto della quasi totale incompatibilità con il software grafico attualmente disponibile. In attesa di avere, ad esempio, un Deluxe Paint 4 con una palette di 16 milioni di colori, Amiga rimane di fatto l'ideale per i sofisticati lavori di titolazione o generazione di logo, il tutto ad un costo veramente basso, considerando che la più piccola titolatrice televisiva professionale (qualche set di caratteri, 8 colori, 2 direzioni di scroll) può venire a costare una decina di milioni.

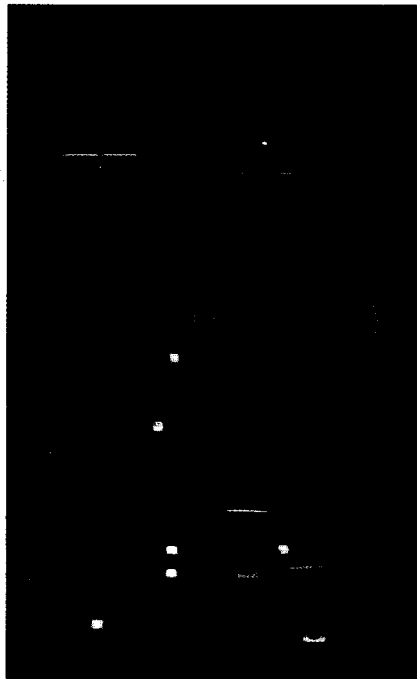
Dopo questa parentesi introduttiva, vediamo concretamente cosa è possibile fare, ed è stato fatto, con un Amiga 2000 presso un network televisivo regionale emiliano-romagnolo: Sestarete. Negli studi di questa rete televisiva, responsabile tra l'altro della diffusione in sede regionale dei programmi di Italia 7 (quelli di Colpo Grosso tanto per intenderci!!!), un sistema basato su Amiga 2000 era già operante da più di un anno per piccoli lavori in ambito pubblicitario. Grazie all'interessamento e l'approvazione del responsabile della rete, Geom. Luigi Ferretti e del notevole sforzo dello staff tecnico, nonché della preziosa e decisiva collaborazione del responsabile delle produzioni Sig. Marco Franzoni, è stata realizzata una cosa ritenuta impossibile da più

Particolare della regia di montaggio dello studio

Un altro particolare della regia di montaggio dello studio

parti: la sigla per il nuovo telegiornale regionale. Nella politica aziendale, per i palinsesti '91 di Rete 8 (l'altra emittente diffusa dagli studi di Sestarete), era in programma la messa in onda del TG come nuovo e importante punto di riferimento per gli avvenimenti di attualità, politica, cultura, sport e spettacolo d'interesse collettivo per l'Emilia Romagna. Si trattava, quindi, di rendere il nuovo TG più incisivo anche dal punto di vista estetico, come ormai siamo abituati, e sottolineare, tramite la computer grafica, le varie rubriche e argomenti di maggior interesse. E' stato deciso di affidare il compito al responsabile per le produzioni Franzoni e al sottoscritto, come collaboratore per la indubbia conoscenza di Amiga e dei pacchetti grafici disponibili (un po' di presunzione non guasta mai), e per una certa vena di creatività, presente in entrambi, che poteva essere così finalmente espressa tra la routine quotidiana dei nostri rispettivi impieghi.

I preparativi

La piattaforma di lavoro, basata su Amiga 2000, era così composta: scheda GVP A3001 (68030/68882 a 28 MHz), 4 Mb RAM a 32 bit, Hard Disk da 80 Mb con tempo d'accesso 19 ms, genlock Magni, digitalizzatore VD Amiga. Inoltre, sono stati utilizzati i seguenti pacchetti grafici: Dpaint 3, DigiWorks 3D 2.0, Pixmate 1.0, VideoScape 2.0, VD Amiga B/W.

Dovrei allungare l'elenco con la lunga serie di attrezzature broadcasting esterne che si sono rese necessarie per la post-produzione di qualità del segnale Amiga (sincronizzazione, correzione, montaggio ecc.), ma si esulerebbe dal fine di questa trattazione che si propone di esporre qualche piccolo suggerimento, spero utile, per la realizzazione di sigle/titolazioni ad uso amatoriale anche su un normale 500

espanso. Procediamo, quindi, con ordine. Innanzitutto è necessario focalizzare quali sono gli elementi grafici (scritte, logo o altro) che dovranno comparire nella sigla. Nel nostro caso erano: l'Emilia Romagna 3D, il logo TG8, il logo Rete8 News e la pagina di un giornale. Dopodiché decidere, a priori, quali colori assegnare a ciascun elemento presente. Quella dei colori meriterebbe un'analisi separata per la complessità dell'argomento che, tra l'altro, richiede: una base di psicologia (su come la mente umana percepisce e da importanza a certi colori), conoscenza delle limitazioni di carattere tecnico del sistema PAL (per evitare sgradevoli accostamenti di certe tonalità cromatiche) e tanti altri fattori che, in ogni caso, per tentativi o inconsciamente ciascuno di noi avrà sperimentato lavorando con programmi grafici. Bisogna, comunque, tener presente che l'output televisivo sarà sempre, anche con sistemi di computer-grafica professionale, differente da ciò che vedete sul vostro monitor. Per l'Emilia Romagna è stato quindi scelto il color verde utilizzando una sfumatura di 8 tonalità, per il logo TG8 una scala sempre di 8 colori sul giallo (per dare l'effetto di color oro), il giornale nella classica scala di grigi per rendere l'immagine meno "piatta" e altri 7 colori sono stati selezionati tra il blu e il ciano per Rete8 News e un motivo geometrico che descriverò più avanti. Abbiamo così completato la palette dei 32 colori disponibili su Amiga. Il colore mancante, il nero come primo colore della palette, è stato tralasciato appositamente perché utile come colore di Chroma-Key per il genlock. Molti sicuramente si chiederanno come mai abbiamo "snobbato" il modo HAM a 4096 colori. Innanzitutto perché anche su un Amiga "turbo" è esasperatamente lento, specie se si vogliono realizzare un migliaio di frame, in secondo luogo per le limitazioni e sbavature che questa modalità comporta. E' stata quindi obbligata (32 colori) la scelta del modo interlacciato (overscan) con una risoluzione di 368x580 pixel. Una definizione sufficiente se gli elementi che compaiono sul video sono realizzati con accortezza e con l'uso dell'anti-aliasing. Per ottenere un buon risultato è parso subito evidente che era necessario usare con sinergia le peculiarità dei differenti pacchetti grafici disponibili. Deciso che la responsabilità di animare il tutto sarebbe toccata all'ottimo DP3, per la flessibilità d'uso, iniziò la fase più importante: la stesura dello story board. Anche se fare degli schizzi su un foglio di carta potrebbe sembrare una procedura superflua, specie se si ha già l'idea in testa, questa fase permette, il più delle volte, di pianificare accuratamente, tenendo in considerazione le limitazioni della macchina, tutte le sequenze che si vogliono realizzare.

Lo Story Board

La sigla richiesta doveva essere lunga al massimo 20 secondi quindi, con 25 frame al secondo, corrispondente a 500 frame. Essendo ogni frame, salvato in formato IFF compattato, lungo in media 40K si giunge ad un totale di 20 Mb d'animazione. Ma non spaventatevi: in realtà il DP3 utilizza uno speciale al-

goritmo che memorizza solamente il delta dei frame (differenza tra un fotogramma e quello precedente) per cui in una animazione fluida, a parte il fotogramma iniziale, si occupano in media (con risoluzione/colori utilizzati) 6K per frame. In definitiva una sigla da 3 Mb è visualizzabile (avendo 4 Mb) con la tecnica in tempo reale detta della RAM Animation. In mancanza di tale memoria si è costretti a suddividere in più parti l'animazione complessiva e montarla in seguito, fisicamente, su nastro. Per comodità e sicurezza (quando si "spreme" software e macchina al limite il guru è sempre in agguato) è stato comunque deciso di realizzare tre parti animate separate da unirsi, alla fine, con il comodo Append del DP3. La prima parte di 4 secondi (100 frame), la seconda di 10 secondi (250 frame) e l'ultima di 6 secondi (150 frame) per un totale di 20 secondi. In realtà sono stati superati abbondantemente i 20 secondi, non per un errato calcolo dei frame, ma perché in PlayAnim il DP3 quando deve scompattare in tempo reale immagini con un delta elevato (scritte che si spostano velocemente o figure particolarmente grandi) non rispetta più di tanto l'opzione dei 25 frame/secondo impostati nel menu d'animazione.Terminata la lunga ma ben pianificata parte creativa e tecnica dello story board, durata, nel nostro caso, un'intera nottata, fu deciso di realizzare una sigla che presentasse quanto segue:

1^Parte:
- Una pagina di giornale.
- La scritta Rete8 News che compare scrollando in rilievo sul giornale a indicare che si tratta di un telegiornale.
- Il "vecchio" media informativo (sempre il nostro giornale) che si lacera a partire dal centro per il "peso" (importanza) di quello televisivo (la scritta Rete8 News).

2^Parte:
- Dietro al giornale lacerato, s'inizia a intravedere una nuova dimensione (nel senso informativo), realizzata graficamente con una specie di "scacchiera" in movimento per dare un effetto di profondità (da associare a maggior completezza delle informazioni, più approfondimento, ecc.).
- Sopra la "scacchiera" vola l'Emilia Romagna, arrivando alle spalle dell'osservatore e sparendo in lontanza con un volo radente sulla "scacchiera" che scorre velocemente (per creare l'associazione dell'informazione "immediata" riferita alla regione).
- Sparisce la scacchiera perché non più visibile dall'osservatore che ora è sopra di essa con lo "sguardo" rivolto verso l'alto.

3^Parte:
- Ritorna l'Emilia Romagna che si fissa nel "vuoto".
- Si forma pezzo per pezzo il logo di TG8, con elementi scrollanti provenienti da differenti direzioni.
- Il logo formatosi in "rilievo" sull'Emilia Romagna rimane qualche secondo con effetti dovuti a riflessioni di luce.
- Mentre tutto sfuma in nero (fadeout) l'osservatore si allontana dal logo.

Qualche considerazione sulla multimedialità

Quanto appena esposto potrebbe essere il pretesto per introdurre un argomento di proporzioni ben più vaste inerente alla multimedialità di Amiga. In questo ambito, comunque, vorrei evidenziare solamente alcune considerazioni di carattere generale. Come al solito, un termine entra a far parte del linguaggio comune perché di moda, infatti la parola multimedia viene erroneamente utilizzata per indicare un "oggetto" in grado di rappresentare interattivamente i risultati, delle proprie elaborazioni provenienti da diverse fonti, sotto forma audio-visiva, o peggio, macchina che serve a varie applicazioni. Anche se, sotto un certo aspetto, questo corrisponde a verità, la realtà è ben diversa e gli attuali multimedia svolgono perlopiù la funzione di sofisticato telecomando. Infatti, avere il computer collegato alla TV, videoregistratore, CD-I, MIDI, digitalizzatore, campionatore, stampante e modem non è certo requisito fondamentale di un sistema multimediale. Al centro di tutto sta sempre l'elaboratore e lo stesso CD-I, da più parti osannato come il multimedia per eccellenza, è in pratica una semplice periferica interattiva (anche se auto-sufficiente) utile, come altri fonti audio-visive, per crearsi una workstation multimediale. Il vero multimedia è un computer capace di elaborare le informazioni audio/visive, provenienti da diverse fonti, e di trasformarle/tradurle in una realtà fisica diversa. Trasformare, ad esempio, musica in immagini, secondo diversi criteri programmati dall'utente o viceversa. Questo sarebbe già uno dei requisiti basilari per dar sfogo alla creatività e realizzare un video-clip in cui immagini digitalizzate da vecchi film di Fred Astaire, musica da CD di genere house, background il ring di un film della serie Rocky vengono fuse in un unico insieme gestito completamente dal computer. Il ballerino, ricolorato dal computer in tempo reale per rispettare luci e ombre, danzerebbe (utilizzando una sequenza apposita di frame per creare movimenti simili a quelli di un ballo moderno) in sincrono con la musica, muovendosi per evitare i pugni dei partecipanti al match. Il ballerino, il filmato di background e la musica, sarebbero completamente gestiti dall'elaboratore che, tramite parametri definiti dall'utente, trasformerebbe la musica in direttive per i movimenti del protagonista e, nello stesso tempo, analizzando le immagini di background, permetterebbe di evitare "collisioni" o eseguire un percorso prestabilito. Molti, giunti a questo punto, avrebbero molte altre idee da aggiungere a questo clip ma, purtroppo, dovremo attendere ancora qualche annetto per poter realizzare questa e tante altre applicazioni multimediali.

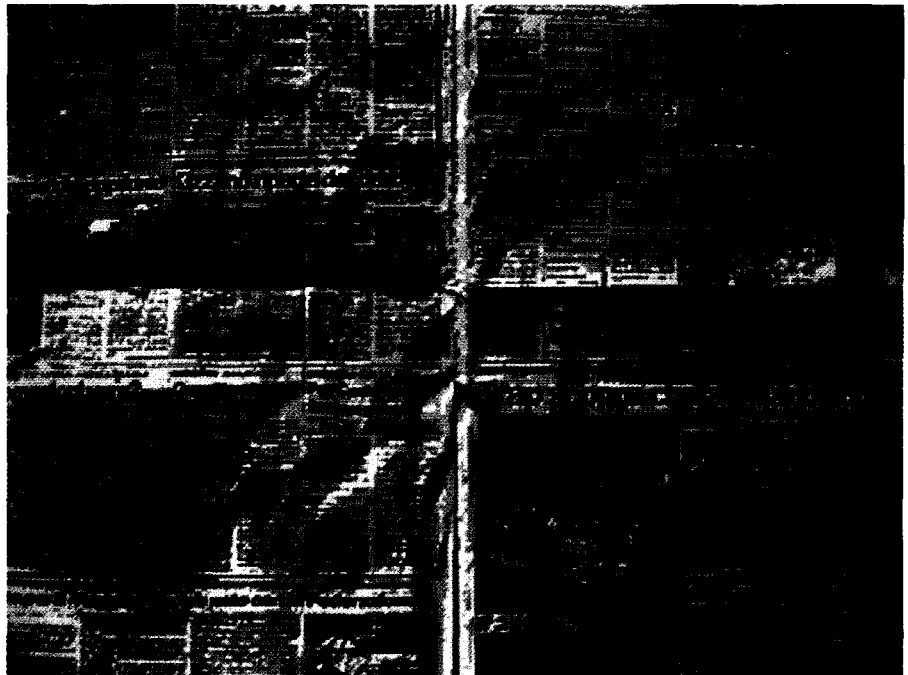
I Pacchetti grafici

L'unica difficoltà nel realizzare animazioni, con elementi grafici provenienti da differenti pacchetti, è quel-

La pagina di giornale digitalizzata

la di assegnare un ben definito range di colori per ciascuno di essi. E' quindi stato deciso di assegnare i primi 8 colori della palette per l'Emilia Romagna, la seconda serie di 8 per il logo TG8, la terza per la scala di grigi e, infine, gli ultimi 7 per la "scacchiera" e logo Rete8 News.

DigiWorks 3D & VideoScape

L'Emilia Romagna è stata digitalizzata da un atlante con il VD Amiga, salvata in IFF, caricata e brushata (scusate il termine, ma spesso sono più chiari delle traduzioni letterali) con l'opzione color di DP3 per ottenere un colore uniforme. L'immagine risultante, risalvata e acquisita con il DigiWorks, diviene così il punto di partenza per la trasformazione da bitmap a vettoriale. Impostate le opzioni per il tipo di superficie (metallico/riflettente), colore (verde), profondità e risoluzione di punti/poligoni, si è ottenuta la versione vettoriale della regione. In seguito è stata salvata e trasformata con il F.O.T. (utility contenuta nel pacchetto VideoScape) per essere utilizzata come object durante l'elaborazione. Il VideoScape, nonostante un certa anzianità, è uno dei pacchetti grafici più sofisticati e intuitivi per le animazioni 3D.
Tant'è vero che, il programma (non in modo HAM e con scheda acceleratrice), selezionate poche opzioni, è in grado di visualizzare l'animazione quasi in tempo reale, mentre esegue l'elaborazione dei frame. Programmate le "luci" e la sequenza di movimenti, sia dell'oggetto che della telecamera simulata (mi si perdoni il termine già sufficientemente usato a sproposito), i vari frame (150) sono stati salvati in formato IFF con lo scopo di poterli rielaborare con il DP3. Caricati con la comoda opzione di Load Picture/Frame# di DP3, facendo attenzione di usare già in partenza la palette d'animazione, sono stati risalvati in formato Anim Brush. La preparazione dell' Emilia Romagna, qui sinteticamente descritta, è stato probabilmente il lavoro più lungo nella realizzazione dell'intera sigla e ha richiesto un decina di ore. Visto il risultato ottenuto ne è valsa comunque la fatica e il tempo speso.

Pixmate & VD Amiga

La pagina di giornale è stata digitalizzata in un pseudo bianco/nero (con una scala di 16 grigi) e in seguito salvata in IFF. L'immagine, così ottenuta e acquisita con il Pixmate, è stata trasformata in soli 8 colori, fissando la palette di grigi in maniera tale da occupare esattamente la posizione spettante in quella d'animazione. Il Pixmate si è rivelato un programma veramente eccezionale, dotato di sofisticati algoritmi che hanno permesso di avere l'immagine a 8 colori praticamente identica a quella a 16 (tenendo anche in considerazione il fatto che i grigi si prestano bene a questo scopo) e di aver egregiamente risolto il problema della posizione in palette dei colori necessari. In seguito l'immagine è stata ritoccata con il DP3 per non far riconoscere la testata e modificata con qualche colpo di brush per modificarne l'impaginazione.

Deluxe Paint 3

Con il più eccellente programmma per animazioni, oltre al "montaggio" finale, è stata realizzata l'animazione dell'enorme "scacchiera". Questa era composta da un modulo formato da quattro quadretti, diagonalmente di colore uguale, brushato e moltiplicato per tutto lo schermo con l'opzione Fill del menu di Move. Le altre impostazioni di questo menu (dist, angle ecc.) e del Perspective, hanno permesso di ottenere l'effetto di volo di picchiata iniziale e planata finale. Siccome il DP3 usa la 2D animation, cioè anche se gli oggetti si muovono sui tre assi, questi rimangono, comunque, figure piatte senza la possibilità di spessore e riflessi di luce, molto cura è stata rivolta alla realizzazione del modulo. Questo è stato disegnato con sfumature blu per un sottomodulo e sfumature grige per l'altro, in moda da dargli in apparenza un aspetto solido (tecnica utilizzata nei disegni animati e nei videogiochi). Le sfumature chiaro/scure sono state disegnate tenendo in considerazione la provenienza dei fasci di luce con cui è stata realizzata l'Emilia Romagna. Il blu è stato scelto per la caratteristica di essere un colore

che, per ragioni tecniche ma soprattutto percettive, possiede un anti-aliasing "naturale" anche quando, durante la sequenza di planata, il modulo di 20x20 pixel assume la dimensione 100x200 o più pixel e per l'ottimo effetto di profondità che si ottiene.

Disegnare un logo

Anche per il logo TG8 è stato utilizzato il DP3 ma questa volta come puro mezzo espressivo grafico. Infatti, nella creazione del logo, finalmente, subentra prepotentemente la componente creativa. Si tratta comunque di una espressione particolarmente intelligente del lato creativo che deve per forza tener presente le caratteristiche tecniche del mezzo su cui opera. Per queste ragioni, spesso, disegnatori professionisti, artisti del pennello e dell'aerografo, esperti in design pubblicitario, sono i primi a fallire miseramente quando hanno a che fare con il computer (ho una notevole esperienza diretta in proposito). Bisogna quindi affidarsi alle proprie capacità o di qualcuno che per diletto ha acquisito una notevole "mano" con programmi di disegno su computer. Bisogna, inoltre, tenere in considerazione lo scopo del logo e il target a cui è rivolto. Nel nostro caso sono state fatte molte considerazioni tra cui queste le principali:

1) Deve essere chiaro (color giallo/oro) ben visibile e soprattutto non molto complesso per essere riconsciuto subito anche fuori dall'ambito televisivo magari stampato (con pochi colori) su adesivi o carta intestatata.

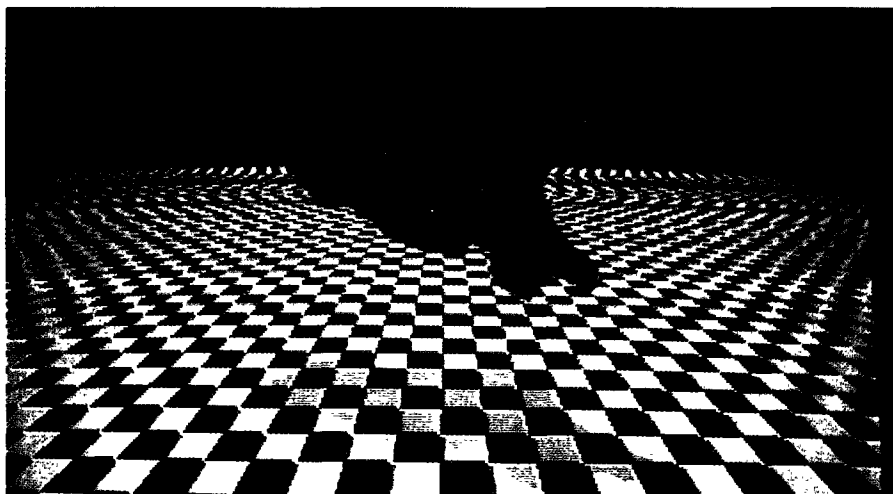
2) Il Telegiornale è seguito da persone di differenti fasce d'età. Quindi, mentre un giovane è abituato a focalizzare immediatamente un prodotto, specie se ha un nome/logotipo sgargiante dai colori vivaci, le persone più avanti con gli anni preferiscono qualcosa di più immediato senza tanti fronzoli.

3) Il Telegiornale è una cosa seria di pubblica utilità e quindi anche il logotipo deve riflettere questa sua caratteristica e non assomigliare, tanto per far un esempio, a quello di un Gel per capelli.

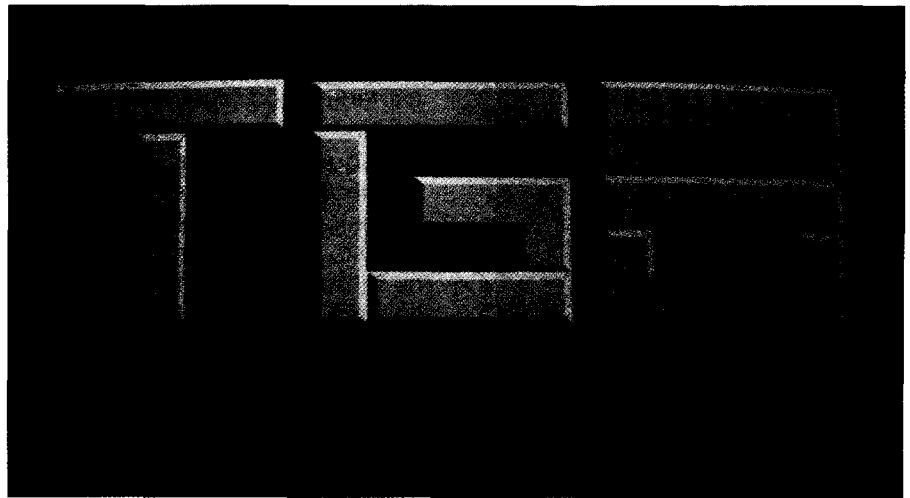
4) Il logo è in genere l'elemento grafico principale di una sigla per cui deve essere studiato in maniera tale che, in fase d'animazione, non venga eccesivamente distorto, perda di risoluzione, cambi di proporzione ecc.

Tenendo in considerazione tutto ciò, è stato costruito il TG8 con un procedimento creativo/analitico, realizzando dei caratteri formati da moduli scomponibili. Questa tecnica ha permesso, in seguito, di formare l'intero logo partendo da una decina di blocchetti grafici (orizzontali e verticali) che si muovevano con apparente casulità sullo schermo fino a formare il TG8.
In maniera analoga è stato realizzato anche il logo Rete8 News ma con maggiore "libertà", essendo questa una scritta introduttiva.

Assembliamo il tutto

Con tutto il materiale "stivato" nell'Hard Disk, fu intrapresa la delicata fase finale: partendo con il giornale come background e impostando i frame a 100 (4 secondi). La scritta Rete8 News entrava scrollando orizzontalmente da 2 direzioni opposte e, una volta fissata al centro, ruotava lacerando il giornale. Effetto ottenuto bloccando i colori della scritta con la funzione Stencil e procedendo manualmente a cancellare (colore di background), frame per frame, con la funzione di Fill di un area definita manualmente, pezzetti di giornale dal centro verso l'esterno. La prima parte così terminata è stata quindi salvata per iniziare la seconda. La "scacchiera" formata da 250 frame (10 secondi), già generata e salvata in precedenza, era pronta per ricevere l'Emilia Romagna in formato Anim Brush da sovraporre. Calcolato accuratamente il punto d'arrivo dell'Emilia Romagna, all'interno dei 250 frame, per far sì che si inclinasse e sparisse all'orizzonte in relazione alla giusta inclinazione della "scacchiera", la seconda parte delle sigla fu subito completata. Ultima parte, la "scacchiera", ora invisibile all'osservatore, fece posto al "ritorno" dell'Emilia Romagna che si avvicinava e si fissava nel vuoto in attesa che, in sovraimpressione, si formasse il logo TG8 per un totale di 150 frame (6 secondi). Tutto queste operazioni sono state fatte con l'uso dell'anti-aliasing e le varie sequenze rifinite manualmente con vari giochi di luce sulle scritte. A questo punto fu deciso di unire la seconda

L'Emilia Romagna che "vola" sopra la "scacchiera"

Il logo TG8 appena formatosi sopra l'Emilia Romagna

e terza parte con il comando Append mentre, per la prima con quella risultante dall'unione delle ultime, si creò il primo vero problema. Infatti, dietro al giornale che si lacerava, si doveva già intravedere la scacchiera, inclinata di 90 gradi, in procinto di iniziare la propria sequenza di movimento. Cosa sufficientemente facile da realizzare, sempre con il DP3, ma lunga e laboriosa giunti a questo punto. Per fortuna il responsabile delle produzioni Franzoni, ha avuto l'idea vincente: registrare su nastro la seconda parte della sigla (quella risultante dalla seconda e terza parte) e genlockarla con la prima prodotta direttamente dal computer. Per le attrezzature professionali usate e per l'ottima resa del Magni, terminato il lavoro, era praticamente impossibile notare delle differenze qualitative. Mi hanno tuttavia spiegato che, in realtà, per via della banda passante, fase del colore, sincronizzazione, risoluzione ecc. c'è una differenza in pratica rilevabile solo strumentalmente. Ovviamente, con il normale VHS di casa (sempre supponendo che ne possediate due per fare questo giochetto), la resa qualitativa sarebbe alquanto diversa per cui, riallacciandomi alla prima parte dell'articolo, consiglio di preparare bene lo story board per evitare di perdere tempo, proprio alla fine, in problemi analoghi. Il "difetto" di flickering, dovuto alla modalità grafica utilizzata, in fase di post-produzione del segnale, scompare completamente per il fatto che i due semiquadri, shiftati tra loro di una linea orizzontale, risultano presenti nello stesso frame. I più smaliziati di voi, esperti nella computer-grafica "amighevole", giunti a questo punto dell'articolo, probabilmente storceranno un po' il naso per la "semplicità" con cui è stata realizzata la sigla. A costoro rispondo, per l'esperienza che ho avuto con diversi utenti, che introdurre tecniche di ray-tracing, Phong e Gouraud shading, disegno poligonale, reflection mapping, texture mapping ecc., avrebbe avuto l'effetto di spiazzare la maggior parte dei lettori il cui interesse è rivolto soprattutto alla realizzazione di una buona sigla per le proprie registrazioni amatoriali o presentazioni/pubblicità per negozi, convegni, fiere e mostre. Il tutto utilizzando pacchetti a basso costo molto diffusi e in tempi accettabili su un normale Amiga 500 espanso. Spero, comunque, di avervi suggerito un valido metodo di lavoro per le vostre prossime realizzazioni (con qualunque configurazione hardware/software). Infine, desidero sottolineare che più delle sofisticate tecniche d'animazione ciò che conta veramente è la creatività.

Conclusioni

Davanti al volto esterrefatto di alcuni operatori, si muoveva fluidamente la sigla sui monitor della post-produzione. Il loro stupore giunse all'apice, con segni d'incredulità, quando rivelammo che il tutto era stato realizzato in meno di 24 ore con un Amiga. E' stata una esperienza utile, con risultati giudicati da più parti eccellenti in relazione alla macchina impiegata, che ci ha insegnato a non sottovalutare il gioiello della Commodore e di quanto, in un prossimo futuro, si potrà fare con un'eventuale nuova scheda grafica. Infatti, sia Amiga 2000 con GVP che il 3000 a 25 Mhz, hanno le stesse prestazioni se non superiori, in termini di velocità e potenza di calcolo, ad analoghi computer grafici basati sui processori Motorola 68020/68030 (per esempio il Matisse) e superiori anche ai tanto strombazzati 386. Ciò che manca, oltre la già citata scheda grafica, è un software che possa sfruttare appieno le nuove risorse di Amiga e soprattutto impostato per un uso professionale. Purtroppo le fotografie pubblicate, a completamento di questo articolo, non rendono giustizia alla qualità della sigla per il fatto che, difficilmente qualche frame estrapolato, da una sequenza in movimento, può dare una corretta idea dell'animazione complessiva. Ah, dimenticavo... anche la musica che accompagna la sigla è stata fatta con Amiga, ma questo è argomento per un altro articolo. ▲

*Stefan Roda è software manager della MultiMedia di Bologna, società conosciuta a livello mondiale, tra gli appassionati del software d'intrattenimento, con il marchio Genias.
Diplomato in elettronica industriale ha programmato su sistemi Z80, 65XX e 68000. Attualmente per l'esperienza acquisita in diversi settori dell'informatica di consumo, coadiuvato da programmatori e specialisti, coordina e supervisiona tutte le fasi, dalla ideazione al packaging, dei prodotti Genias.*

I Codici di Escape

Jim Butterfield

Anche questa mese alcuni consigli veloci: si tenti il comando PROMPT "*e[33m %S *e[0m *n> ". Effettuerà dei cambiamenti interessanti nel prompt della Shell o del CLI. La directory corrente sarà visualizzata in arancio e il carattere di prompt (un segno "maggiore-di", ma si può cambiare, se lo si desidera) apparirà sulla linea successiva. Si potrebbe trovare utile questa funzione se si ha un path molto lungo, che occupi buona parte della linea CLI.

Come funziona

PROMPT è, naturalmente, il comando che modifica il messaggio di prompt che si vede sulla linea di comando, il resto della linea è racchiuso tra virgolette. Queste virgolette fanno di più che permettere di scrivere degli spazi entro la stringa. Rendono disponibili delle speciali funzioni che vengono abilitate dall'asterisco.
Entro le virgolette, l'asterisco si trasforma in un codice di escape che segnala che segue qualcosa di inusuale. Le combinazioni più importanti sono *n, che rappresenta il carattere di newline; *e che rappresenta il carattere di escape; e la combinazione *" e **, che generano le virgolette e l'asterisco.
La parte dei dati nella nostra stringa di prompt è *e[33m (attenzione, la lettera "m" deve essere in minuscolo!). Questo fa diventare arancio il colore di stampa. I primi tre caratteri, *e[, costituiscono il cosiddetto CSI o Control Sequence Introducer. Il CSI segnala che seguirà una sequenza di controllo (dati numerici seguiti da un carattere alfabetico). La lettera m è il comando Set Graphic Rendition e il codice 33 selezionerà il colore arancio.
Continuando con il nostro esempio, troviamo %S. Come è documentato nel manuale dell'1.3, esso genera la visualizzazione della directory corrente nel prompt. Dopodiché troviamo *e[0m. Adesso si può riconoscere in *e[il CSI e in m il comando Set Graphic Rendition. Il valore 0 fa ritornare tutto normale; è un modo generico per ritornare alle condizioni di default. In questo caso il colore di stampa torna ad essere il bianco.
Proseguendo, arriviamo al segnale *n che indica un newline. Infine, abbiamo il carattere di prompt, >, e uno spazio finale.
L'intera sequenza potrebbe essere espressa come "seleziona l'arancio, stampa il path della directory, torna ai colori normali, vai alla linea seguente e stampa i caratteri di prompt". Un notevole lavoro per il comando, semplice in apparenza, PROMPT.

Impostazione della stampante

Di tanto in tanto si ha bisogno di stampare un documento con la stampante. Ci sono molti modi per farlo. Un file può essere stampato con un semplice comando come COPY FILE PRT: o TYPE >PRT: FILE. Se ciò che si desidera stampare è l'output di un programma, si può spesso usare la ridirezione. Il comando PROGRAM >PRT: dirigerà l'output di PROGRAM verso la stampante invece che verso lo schermo.
Ecco un altro metodo: in molti programmi di text editor, o anche in AmigaBasic, si può chiedere di salvare il documento come file di testo. Se si fornisce PRT: come nome di file, il vostro documento andrà verso la stampante invece che verso un file su disco. Per esempio, se si sta lavorando in AmigaBasic e si desidera il listato di un programma, SAVE "PRT:",A o LIST ,"PRT:" faranno una copia del programma proprio come fa, più semplicemente, LLIST.
A me piace aggiungere un titolo ai documenti, fornendo un nome e magari una data, aiuta a tenere in ordine i vari fogli. Qualche volta dirigo l'output in un file nel RAM Disk, lo edito e poi invio il documento "abbellito" alla stampante. Ma è spesso più comodo spedire direttamente alla stampante un breve titolo prima di inviarvi il testo principale. Lo si faccia digitando ECHO >PRT: "Titolo, data e altro ancora". Stampata la linea, possono seguire il file o altre informazioni.
Ora che conosciamo CSI, possiamo fare molte altre cose. Le tabelle che accompagnano l'articolo mostrano alcuni di questi comandi. Io amo particolarmente usare le seguenti funzioni, quando sono appropriate: corsivo, sottolineato, neretto e stampa compressa o espansa. Alle volte ho la necessità di selezionare i font nazionali: Francese, Danese, Spagnolo o Inglese, magari.
Si tenti di stampare in corsivo sullo schermo digitando ECHO "Questo è *[3m molto *e[0m interessante!". Poi si prosegua inviando il corsivo verso la stampante con ECHO >PRT: "Questo è *[3m molto *e[0m interessante!".
Lo stesso codice produce gli stessi effetti su dispositivi di output molto diversi. Si può fare la stessa cosa con i modi sottolineato e neretto. In

ciò emerge il modo meraviglioso con cui Amiga gestisce i codici di stampa: sebbene ogni stampante potrebbe avere bisogno di ricevere una sequenza di codici diversa per generare, ad esempio, il corsivo, con Amiga si deve stampare la medesima sequenza, senza riguardo al tipo di stampante che si sta usando!
I dati inviati al device PRT: vengono tradotti nei codici dedicati adeguati dal driver della stampante che è stato selezionato in Preferences. Una volta selezionato il driver di stampante adatto, tutti i programmi che usano PRT: vedranno i loro dati adattarsi alla stampante usata. Si noti che ciò vale solo per il device PRT:, se si usa direttamente PAR: (o, per una stampante seriale, SER:) i dati non verranno modificati.
Si controllino le possibilità della stampante, provando alcuni dei comandi che seguono:

```
ECHO >PRT: "*e[2w Elite! *e[0w"
ECHO >PRT: "*e[4w Compressa! *e[0w"
ECHO >PRT: "*e[6w Espansa! *e[0w"
ECHO >PRT: "*e[4*"z Test della stampa in double-strike! *e[3"z"
ECHO >PRT: "*e[2*"z Test della stampa in NLQ! *e[1"z"
ECHO >PRT: "*e(Z #[!]? *e9B"
```

Negli esempi, sono stato attento a cancellare l'effetto selezionato. Ogni opzione viene abilitata, testata e poi disabilitata. Non si deve per forza fare così.

Upgrade del Workbench 1.3

La Commodore ha rilasciato una nuova versione del Workbench 1.3, numerata 1.3.2. I mutamenti rispetto al sistema 1.3 originale sono principalmente rivolti agli utenti di hard disk. La revisione 1.3.2 è apparsa originariamente, nel momento in cui la Commodore l'ha resa disponibile, come modulo scaricabile da certe BBS. E' accompagnata da una licenza che consente ai singoli utenti di scaricare il pacchetto, ma ne proibisce una distribuzione ulteriore in qualsiasi forma.
C'era una certa curiosità nella comunità Amiga sul perché la Commodore fosse stata così generosa verso la distribuzione elettronica a vasto raggio e ancora così apparentemente avara nel restringere le possibilità di distribuzione ulteriore. La ragione: in questi tempi di virus, la compagnia desiderava avere un sentiero ben delineato fra sé e gli utenti. Ulteriori mediazioni avrebbero potuto aprire le porte ad eventuali abusi. Ci si aspetta che i rivenditori autorizzati Commodore abbiano a disposizione la corrente versione del Workbench 1.3.2 per coloro (specie utenti di hard disk) che ne hanno bisogno. Le precedenti versioni del Workbench sono ancora utilizzabili, sebbene si dica che la versione 1.3.1 del comando DISKCOPY contenga dei bug.

Hard disk e floppy disk

Tutti i sistemi Amiga, senza riguardo all'età e alla configurazione, hanno molto in comune. Ma ci possono essere delle differenze nel modo in cui gli utenti di hard disk e quelli di floppy disk organizzano e salvaguardano il loro sistema di memorizzazione.
Gli utenti che si basano sui floppy da 3.5" tendono a creare un insieme di dischi "dedicati", ognuno dei quali è confezionato per lanciare il sistema in un determinato modo. Ci potrebbe essere un disco per il word processor, uno per il disegno grafico e un altro per la programmazione in AmigaBasic, per esempio. Per risparmiare lo spazio, l'utente pianificherà attentamente il contenuto di ogni disco. Il drawer Preferences verrà spesso rimosso da tutti i dischi di questo tipo, una volta impostati i colori e selezionata la stampante appropriata. La maggior parte del contenuto del drawer Utilities sarà eliminata, assieme ad alcuni degli elementi meno usati nel drawer System. E' probabile che il disco usato per la programmazione in Basic perda i font aggiuntivi. Dal disco per la grafica probabilmente verranno rimossi gli elementi per la sintesi vocale, LIBS:TRANSLATOR. LIBRARY e DEVS:SPEAK-HANDLER.
Ogni disco conterrà una startup-sequence dedicata. Una parte importante della configurazione consisterà nell'uso del comando ASSIGN per identificare le aree speciali usate da ciascun applicativo. Un word processor potrebbe aver bisogno che si assegni il path in cui trovare il dizionario; un compilatore potrebbe aver bisogno del path per i file include. A questo modo, l'utente con floppy disk personalizza il sistema in fase di startup ad ogni boot. L'utente di hard disk, da parte

Codici di controllo

*n	Newline
*ec	Reset stampante / pulisce schermo

Comandi CSI m

*e[1m	Neretto
*e[3m	Corsivo
*e[4m	Sottolineato
*e[32m	Colore di foreground nero
*e[33m	Colore di foreground arancio
*e[42m	Colore di background nero
*e[0m	Ristabilisce i valori di default
*e[32;41m	Foreground nero, background bianco

Comandi CSI w

*e[2w	Elite
*e[4w	Stampa compressa
*e[6w	Stampa espansa
*e[0w	Ristabilisce la stampa normale

Comandi CSI "z

*e[*4"z	Stampa in double-strike
*e[*3"z	Elimina double-strike
*e[*2"z	Stampa in NLQ
*e[*1"z	Elimina NLQ

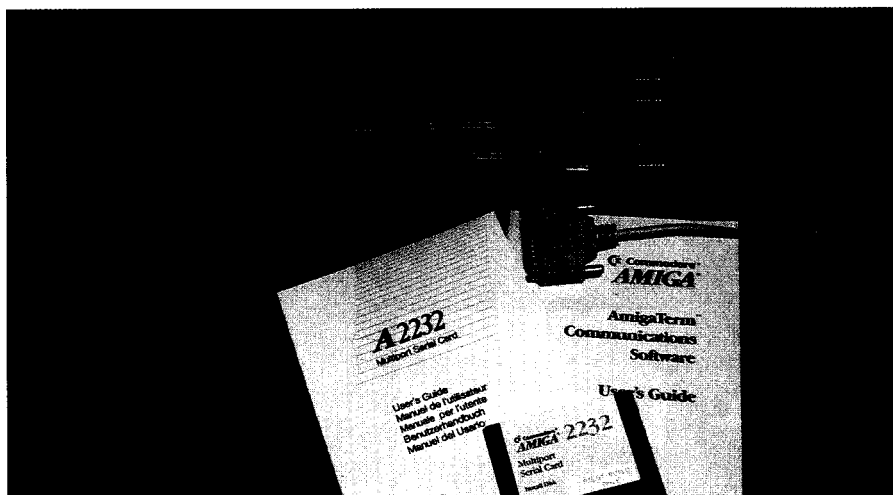
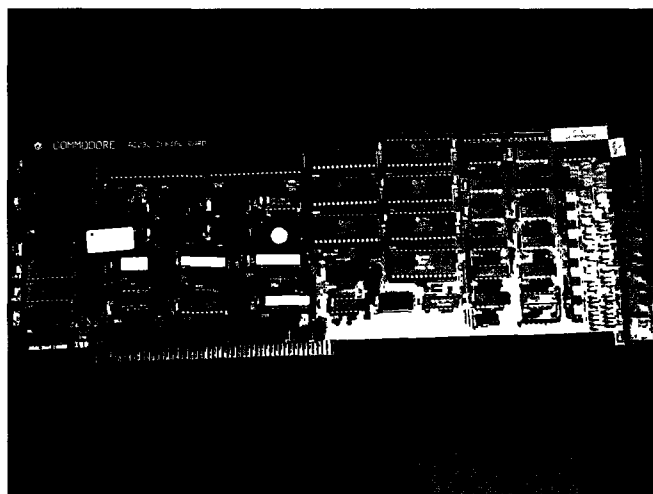
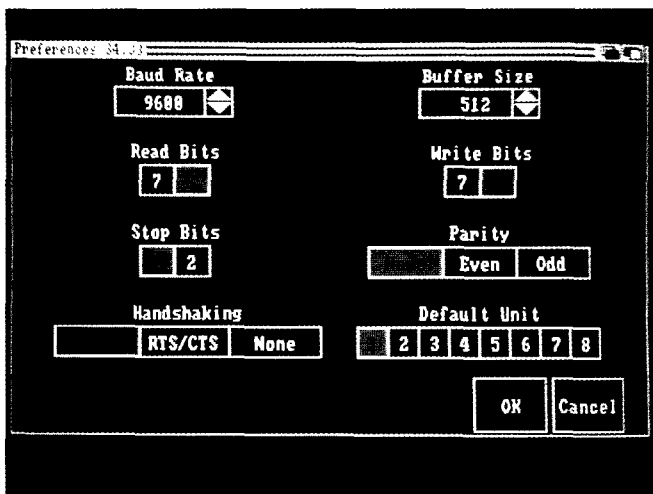
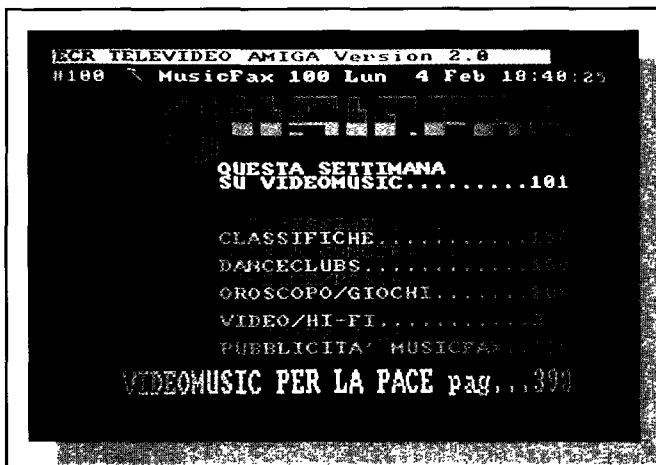
Comandi CSI per i caratteri nazionali

*e(B	USA
*e(R	Francia
*e(A	Regno Unito
*e(E	Danimarca I
*e(Z	Spagna
*e(Y	Italia

sua, può caricare tutto allo stesso tempo. Avrà a disposizione una collezione gigante di font, più tutte le librerie aggiuntive che potrebbero essere utili ai vari programmi. Tutti i comandi ASSIGN che potrebbero essere utili vengono effettuati nella startup-sequence.
Sembra comodo, niente più caccia a un determinato floppy prima di iniziare un lavoro, ma presenta alcuni piccoli inconvenienti.
I maggiori problemi degli utenti di hard disk sono il disordine e l'integrità dei file.
Il disordine è un problema che si presenta di soppiatto dietro le spalle dell'utente di hard disk.
C'è così tanto spazio su un hard disk tipico che sembra non ci sia bisogno di liberarsi dei vecchi materiali. Il file vecchio, indesiderato, rimane sul disco, ma si dissolve dalla memoria dell'utente. Quando il disco finalmente comincia ad affollarsi (sì, anche un'unità da 40 Mb si riempie ad un certo momento), l'utente spesso non ricorda a cosa serviva la maggior parte di quei vecchi programmi e file.
L'integrità dei file è un problema dalla natura simile.
Gli hard disk sono molto affidabili e spesso funzionano per anni senza problemi.
Ma quando il problema si presenta, il disco può perdere un grande numero di programmi e di file di dati. Può risultare difficile ricostruire un paio di centinaia di programmi perduti, è virtualmente impossibile ricostruire i file di dati perduti.
E' necessario pianificare delle procedure di backup, che vanno rispettate scrupolosamente.
Questo sarà d'aiuto sia al problema dell'integrità dei file, in quanto i file possono essere recuperati se sono andati perduti, sia i problemi di disordine, dal momento che si possono eliminare in maniera relativamente sicura i file inutili, nella certezza che è possibile recuperarli.
L'utente senza hard disk può vedere il computer in modo diverso, ma il problema esiste ugualmente. La libreria dell'utente può rapidamente crescere fino a raggiungere il numero di centinaia di floppy.
L'eventualità di un bad disk che causi la perdita di dati o programmi va ancora tenuta presente. Per l'utente di floppy disk un sistema di archiviazione è ancora importante. Si facciano dei backup o si faccia molta attenzione ai dischi vecchi.▲

Scheda Multi-Port A2232

Stefano Paganini

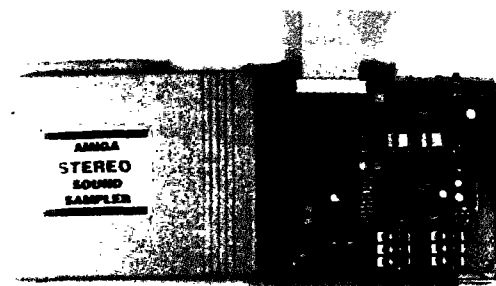
L'introduzione del modello 2000, ha costituito la vera e propria risposta alla richiesta di espandibilità degli utenti Amiga e il recente 3000 completa questa risposta anche in termini di potenza di elaborazione.
Come è logico, oltre ai nomi già coinvolti nel mercato hardware per Amiga, la stessa Commodore ha provveduto alla dotazione di schede aggiuntive in modo da rendere il sistema sempre più potente, flessibile e versatile.
La scheda in prova questo mese, rappresenta un esempio tipico di questa politica: si tratta della scheda Multiporte Seriali A2232.
Cominciamo dalla descrizione della scheda che, occupando uno slot qualunque, fornisce il 2000, di ben sette porte seriali RS232 completamente configurabili, cui si aggiunge la porta standard del 2000.
La scheda è basata su un 6502 cui è affiancato un UART 8520, e dispone di una RAM dual port di 16K della quale il manuale fornito in dotazione fornisce locazioni e dati sui registri delle singole porte seriali.
E' curioso notare come il 6502 e i suoi derivati vengano utilizzati come coprocessori in più macchine: anche il potente Mac IIfx utilizza due 6502 a 10 MHz in versione surface mount.
Vicino ai convertitori di livello TTL/RS 232, i noti 1488/1489, vi sono i ponticelli per la configurazione delle singole porte, DTE/DCE.
Sul retro della scheda sono presenti 7 connettori a 7 poli, simili al connettore standard Apple, cui vanno collegati i cavi, di ottima qualità, forniti insieme alla scheda, che terminano con il classico connettore a 25 poli

standard. Più in generale, la A2232 è realizzata con una qualità in linea con il nome che la garantisce.
L'installazione della scheda è quantomai semplice e, come già accennato, è da notare che la documentazione inclusa è completa e comprende una sezione, dedicata alla programmazione della scheda, che va ad integrare le informazioni dei vari Reference Manual ufficiali della Commodore.
Vengono forniti esempi sull'utilizzo del serial.device ampliato che si trova sul dischetto fornito con la scheda, sulla nuova Mountlist e sul nuovo pannello Preferences che, nella schermata dedicata Change Serial, dispone di un gadget aggiuntivo per la selezione della porta seriale di default.
La documentazione risulta molto chiara anche nella fase di configurazione delle singole porte e della scelta del protocollo di handshaking con chiari schemi di collegamento (null modem, DTE/DCE).
Sul dischetto si trova anche il programma AmigaTerm versione 1.2i che, oltre alle normali funzioni del programma terminale, permette di selezionare la porta seriale in uso.
Questa versione del pacchetto prevede anche alcune migliorie, quali la gestione di font aggiuntivi, incluse nel dischetto, per l'emulazione grafica dei terminali DEC.
Per quanto concerne le prove effettuate, dobbiamo far notare che per il momento il software in grado di gestire questa scheda, e le sue potenzialità, praticamente non esiste.
E' quindi quantomai apprezzata l'esauriente spiegazione del manuale.
Abbiamo effettuato varie prove sia da terminale, AmigaTerm, sia modificando programmi, in C e Basic, in modo da passare da una porta all'altra: in nessun caso abbiamo riscontrato problemi di sorta anche utilizzando la scheda alla massima velocità di trasmissione con protocolli di pacchetto tipo XMODEM/YMODEM/ZMODEM, tra i più utilizzati. E' il manuale stesso a segnalarci che la scheda non può

funzionare come seriale MIDI, in quanto l'hardware non può generare il protocollo di trasmissione e il baud rate dello standard MIDI, cui comunque supplisce la seriale interna.
In prospettiva, i problemi per questo prodotto sono legati al tipo di utenza interessata e alla disponibilità di pacchetti applicativi.
L'utente tipico della A2232 ha una stazione 2000 o meglio 2500/3000, sotto UNIX, e il campo di applicazione può spaziare dalla BBS multilinea, vi sono parecchi programmi USA di questo tipo, al vero e proprio ambiente multiuser, subordinato ad una CPU 020 o 030 e ad un software dedicato oppure al controllo di strumentazione o di processo.
Si tratta sicuramente di un prodotto mirato per un'utenza professionale e dal prezzo tutto sommato giustificabile, cui tuttavia dovrebbe corrispondere un impegno sul fronte del software per impedire che il prodotto passi, ingiustificatamente, inosservato. ▲

Campioniamo i Campionatori Audio

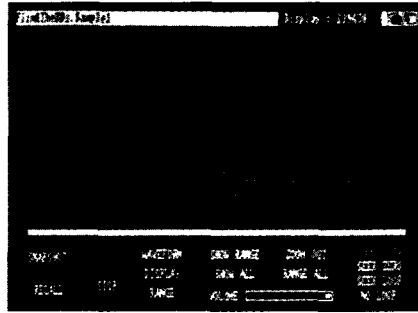
Morton A. Kelvenson

Sebbene le qualità grafiche di Amiga abbiano monopolizzato l'attenzione, le sue qualità audio non devono essere sottovalutate. Amiga ha la bellezza di quattro canali di conversione digitale/analogico a 8-bit che terminano ad un connettore stereo analogo a quelli che si trovano in molti componenti audio. C'è una vasta gamma di applicazioni per i suoni di Amiga. L'uso più comune, è quello dei suoni campionati nei giochi. La multimedialità potrebbe valorizzare radicalmente i suoni campionati. Questo tipo di applicazione si aspetta da Amiga qualità audio, oltre che qualità video. Sebbene Amiga possa riprodurre suoni digitalizzati senza alcuna assistenza esterna, oltre a quella di un amplificatore e di un altoparlante, non è in grado, da solo, di registrare suoni campionati. La soluzione viene dall'aggiunta di un dispositivo hardware noto come campionatore o digitalizzatore audio, che consente ad Amiga di "ascoltare" suoni reali o il loro equivalente elettronico. Fortunatamente, l'avvento di chip di conversione analogico/digitale (A/D) ad alta potenza a basso costo, ha reso i digitalizzatori audio semplici ed economici. Come accade per altre fonti di dati digitali grezzi, ciò che si ricava all'inizio non è spesso ciò che si desidera realmente. Ciò crea una nicchia per il software predisposto alla fase di editing dei dati campionati. Ci sono innumerevoli campionatori e programmi di editing dei suoni sul mercato. In questo articolo ne prenderemo in considerazione una buona parte. Sono inclusi dettagli tecnici per coloro che sono interessati all'argomento, ma anche se non si capisce la tensione di picco di "Twins Peaks", si troveranno in questo panorama molte informazioni utili.

Perfect Sound

Perfect Sound 3.1 è un pacchetto composto dal software di elaborazione dei suoni e un digitalizzatore chiuso in un cabinet di plastica. Quest'ultimo, dalle dimensioni di un mazzo di carte, si collega alla porta parallela di Amiga.
Due connettori audio del tipo RCA accettano un segnale audio stereo. Un piccolo jack da 3.5 mm accetta un segnale mono proveniente da un microfono. La sensibilità dell'ingresso può essere variata attraverso un controllo digitale a 16 livelli dall'interno del programma di elaborazione del suono. L'hardware del campionatore si fonda su un singolo chip di conversione analogico/digitale AD7575JN.
Perfect Sound è in grado di digitalizzare un segnale mono alla velocità massima di 40000 campioni al secondo. Tuttavia, l'esecuzione rimane sempre limitata dalla velocità di campionamento interna di Amiga, pari a circa 28000 campioni al secondo. Il campionamento stereo si limita a una velocità massima di 14000 campioni al secondo.
Si può applicare ai campioni un filtro passa-basso; tuttavia, il processo richiederà alcuni minuti con i campioni più lunghi.
I campioni possono essere ricampionati con una nuova velocità di campionamento senza mutarne l'altezza.
Si possono portare nella tonalità desiderata combinando il ricampionamento con un cambiamento nella velocità di campionamento in fase di esecuzione. Perfect Sound consente di moltiplicare o dividere i campioni di un fattore di 2, per creare le ottave addizionali necessarie a raggiungere le tre o cinque ottave proprie agli strumenti IFF. Permette anche di decidere il punto di looping in uno strumento IFF. Di fatto qualsiasi insieme di tre o cinque campioni può essere combinato in uno strumento IFF. La parte superiore dello schermo di Perfect Sound è una rappresentazione grafica del campione corrente. Si possono definire dei campi in questa finestra semplicemente selezionando e trascinando il puntatore. L'ampiezza dei campioni di Perfect Sound è limitata solo dalla memoria disponibile, che comprende la FAST RAM oltre alla CHIP RAM. Tuttavia, certe operazioni sono limitate a campioni che stanno interamente in CHIP RAM. Si può elaborare dei campioni stereo, ma solo un campione apparirà nella finestra. Sebbene Perfect Sound non limiti la lunghezza del campione, non permette di lavorare con più

Lo Stereo Sound Sampler della Datel presenta ingressi per microfono e altri componenti audio

L'AudioMaster III della Oxxi/Aegis

di sei campioni contemporaneamente. Non si tratta di un numero troppo vasto, in quanto molte delle funzioni di elaborazione generano come risultato un nuovo campione. Per esempio, la creazione di uno strumento IFF a cinque ottave richiede l'uso di 5 campioni, uno per ottava. L'ultima versione di Perfect Sound comprende l'eco in tempo reale e funzioni di ritardo. Il segnale viene accettato dal canale sinistro, viene elaborato e, infine, posto in uscita sul canale destro. Si tratta di una funzione che lavora solo in tempo reale, i dati non possono

essere salvati. Perfect Sound è stato uno dei primi campionatori sonori per Amiga.
E' stato usato da molti sviluppatori per gli effetti sonori dei giochi e l'hardware è ben supportato da programmi di elaborazione di altre compagnie.

Due programmi europei

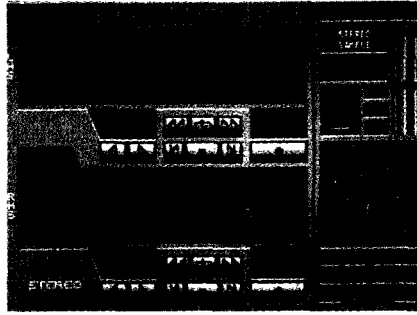
Mi è capitato di parlare di questo articolo durante una recente conversazione con il Dr. Oxide della Pulsar, ed egli mi ha immediatamente fornito due dei suoi ultimi prodotti d'importazione: Real-Time Sound Processor e Digital Studio. Entrambi i programmi sono prodotti dalla Adept Development. Real-Time Sound Processor accetta campioni audio provenienti da un digitalizzatore audio mono o da un canale di un digitalizzatore stereo. I campioni vengono elaborati in tempo reale e inviati alle uscite audio di Amiga. Siccome il manuale, tradotto in inglese, è del tutto incomprensibile, mi ci sono voluti un po' di esperi-

Uno sguardo ravvicinato ai campioni audio

Amiga usa un chip custom per leggere byte di dati e convertirli in tensione analogica. Il chip custom può effettuare questa operazione 28687 volte al secondo per ciascuno dei quattro canali. E' la stessa tecnologia usata dai lettori CD per generare quel suono di alta qualità cui molti di noi si sono abituati. La differenza sta nel fatto che la tecnologia dei lettori CD legge i dati digitali due byte alla volta, 44000 volte al secondo per ciascuno dei due canali. Questo significa che i lettori CD hanno una risposta in frequenza reale e in alta fedeltà di 20000 vibrazioni al secondo e un rapporto segnale/rumore di 90 decibel, circa 65000 a 1. In confronto Amiga raggiunge le 14000 vibrazioni al secondo e un rapporto segnale/rumore di 256 a 1. Sebbene questi numeri non sembrino così impressionanti, sono meglio di quanto vi potreste aspettare. La qualità del suono di Amiga è almeno pari a quella di un buon registratore a cassette, e in molti casi la sorpassa. Di fatto, è possibile ottenere risultati puliti e significativi con velocità di campionamento molto inferiori al massimo di cui è capace Amiga. La maggior parte dei suoni che si ascoltano su Amiga vengono eseguiti ad una velocità di campionamento minore di 10000 campioni al secondo. Amiga ha anche un filtro elettronico built-in che limita la risposta in frequenza a un massimo di 4000-7000 vibrazioni al secondo. Su Amiga 1000 questo filtro è abilitato in modo permanente. Su tutti gli ultimi Amiga questo filtro elettronico può essere disabilitato via software. C'è una ragione molto buona per l'inclusione del filtro. A 28000 campioni al secondo si riempie un Megabyte di memoria con circa 30 secondi di suoni digitalizzati. Per la maggior parte delle applicazioni la limitazione delle frequenze, che si traduce in una velocità di campionamento più bassa, è del tutto adeguata. Il filtro elettronico è necessario per eliminare la distorsione aliasing che diventa un problema alle velocità di campionamento più basse. La distorsione aliasing è un inevitabile sottoprodotto del processo di esecuzione digitale. La banda audio è limitata alla metà della velocità di campionamento. Quando un suono campionato viene riprodotto, il suono originale viene rigenerato assieme a delle immagini del suono stesso la cui tonalità è uguale alla differenza tra la somma della frequenza del suono originale e la frequenza di campionamento. Per esempio, se si esegue un tono da 1000 vibrazioni al secondo con una velocità di campionamento di 5000 campioni al secondo, si udirà il tono originale più toni addizionali di 4000 (la differenza tra 5000 e 1000) e 6000 (la somma di 5000 e 1000) vibrazioni al secondo. Dal momento che i suoni reali sono in realtà combinazioni complesse di un vasto insieme di vibrazioni, ciò che si sente è il suono originale più un paio di copie in alta frequenza. La maggior parte delle persone descrive l'aliasing (o "birdies", uccellini) come un irritante fruscio che accompagna il suono originale. Un mezzo per eliminare la distorsione aliasing, è quello di campionare a una velocità molto alta. La velocità di 44000 campioni al secondo, usata per la registrazione dei CD, assicura che la distorsione aliasing si trovi sempre al di sopra di frequenze di 24000 vibrazioni al secondo. Questa frequenza è ben sopra la banda dell'udibile. Siccome Amiga non usa una tale velocità di campionamento, un circuito che filtri il segnale in uscita è l'unico modo per eliminare la distorsione aliasing prodotta alle basse velocità di campionamento.
Limitare la banda del suono originale fa poco per alleviare il problema. Il filtro built-in di Amiga comincia da attenuare il segnale a 4000 vibrazioni al secondo ed elimina virtualmente qualsiasi segnale al disopra delle 7000 vibrazioni al secondo. Funziona bene con velocità di campionamento quali 8000 campioni al secondo: la maggior parte dei programmi musicali per Amiga presume che uno strumento musicale digitalizzato suoni il do centrale a una velocità di campionamento di 8363 campioni al secondo.

L'editor stereo della Datel

menti per capire cosa può fare il programma. Real-Time Sound Processor permette di controllare e modulare l'ampiezza (AM) e la frequenza (FM) dei dati campionati. Fornisce un completo controllo dell'intensità e della frequenza di modulazione, oltre a consentire la scelta fra quattro diversi inviluppi. Si possono applicare simultaneamente la AM e la FM, in diversi gradi, a entrambi i canali. In più il canale B può essere ritardato e invertito per creare una vasta gamma di effetti di eco e riverbero. Il programma può anche suonare un campione alla rovescia sul canale B, mentre lo esegue normalmente sul canale A. Il modo migliore per imparare ad usare Real-Time Sound Processor, è quello di studiare le sue 20 configurazioni built-in. Esse comprendono effetti come eco, reverse, ping-pong, delay, vibrato, flanger, chorus, e quattro modi per produrre suoni vocali generati dal computer o tipo robot. Compresi gli effetti built-in, il programma può memorizzare fino a un massimo di 128 configurazioni dell'utente. Gli effetti possono essere attivati mediante un'interfaccia MIDI. Il programma consente di salvare effetti su disco, ma, a causa della povertà della documentazione, non sono stato capace di capire come funzionasse. Digital Studio consente di lavorare con dieci suoni campionati simultaneamente. Dispone delle usuali operazioni di elaborazione, come taglia, copia, incolla, rovescia, aggiungi e sottrai, assieme ad eco, rethuning e resampling. Le operazioni si possono effettuare su uno o su entrambi i canali. Si possono salvare i campioni come file IFF, come dati grezzi e in formato compresso. Si possono anche salvare strumenti IFF a tre o cinque ottave. I diversi campioni possono essere combinati fra loro in

Il QuasarSound della Centaur

sequenze e poi eseguiti automaticamente. I file di sequenze usano un formato proprio a Digital Studio che non è compatibile con il formato IFF usato da AudioMaster III. Un programma dimostrativo che si autoesegue, incluso nel disco, mostra con efficacia quello che Digital Studio può fare. Digital Studio sembra supportare solo l'hardware di Perfect Sound 2.0 e di A.M.A.S.

AudioMaster

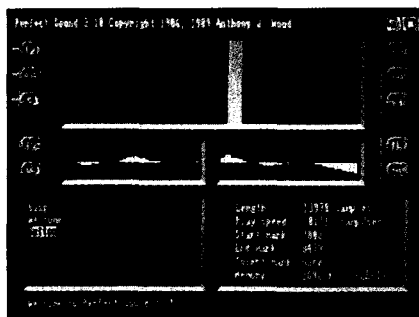
Una beta version della terza incarnazione di AudioMaster, è arrivata appena in tempo per l'inclusione in quest'articolo. Come editor di suoni digitalizzati stand-alone (senza hardware), AudioMaster ha un'interfaccia utente che giudico più curata e intuitiva di quella dei programmi che accompagnano i pacchetti hardware. AudioMaster III può campionare ed elaborare suoni mono e stereo. Può velocemente riversare un campione stereo in un singolo canale o spezzare un campione mono in due suoni identici per il canale di destra e di sinistra. AudioMaster III è il solo editor che supporta il formato IFF per i campioni stereo. Di fatto, il formato IFF per i suoni stereo è stato usato per la prima

volta da AudioMaster II. Nello sforzo di superare se stesso, AudioMaster III introduce ancora un altro formato IFF per i file sonori: il loop multiplo o sequenza. Tale formato di file è in grado di memorizzare fino a 999 loop o frasi, e consente di eseguire ripetutamente differenti porzioni del campione sonoro in qualsiasi ordine. Permette anche di scegliere un punto di dissolvenza in modo che il suono diminuisca fino al silenzio, dopo che un particolare loop sia stato eseguito. Il file può essere usato per salvare intere composizioni musicali in un solo file, che occupa poco spazio su disco. Le sequenze sono facili da creare e affascinanti da vedere in AudioMaster III. Il cursore su schermo e il marcatore dei loop saltano sul grafico del campione sonoro man mano che la sequenza viene eseguita. I digitalizzatori hardware supportati da AudioMaster III comprendono tutte le versioni di Perfect Sound, A.M.A.S. della Mimetics Michtron, StarSound e FutureSound 500. C'è anche un'opzione per digitalizzatori ad alta velocità che potranno apparire in futuro che supporta velocità di campionamento pari a 38080 campioni al secondo in stereo e 55930 campioni al secondo in mono. Le velocità di campionamento dipendono dall'hardware e partono da un minimo di 14914 campioni al secondo per il digitalizzatore della Mimetics che si collega alla porta joystick, per arrivare a un massimo di 23243 campioni al secondo in mono e 55930 campioni al secondo in stereo con A.M.A.S. La velocità di esecuzione può raggiungere i 55930 per secondo in qualsiasi modo. Questi valori si riferiscono ad un Amiga standard con 68000 a 7 MHz. Velocità superiori possono essere raggiunte con alcuni campionatori su sistemi dotati di 68020 o 68030, sebbene la massima velocità di campionamento rimanga comunque di 55930 campioni al secondo. L'AudioMaster III fornisce eccellenti funzioni per la creazione di strumenti. Gli strumenti a una, tre o cinque ottave possono essere creati automaticamente e salvati sia

Il Perfect Sound 3.1 della SunRize

in file IFF standard che in formato Sonix. Una funzione speciale di tuning permette di modificare il campione base per farlo suonare in Do centrale a 8363 campioni al secondo. C'è anche un modo hi-fi che permette di usare per i file degli strumenti a più ottave campioni con velocità superiori a 8363 campioni al secondo. Sono anche presenti: eco in tempo reale, ritardo ed effetti flange, che non producono dati salvabili. Il segnale originale è inviato a uno dei canali audio di Amiga, mentre il segnale elaborato è inviato all'altro canale. AudioMaster III è accompagnato da un pratico programma di esecuzione a più campioni, che sembra e funziona come un lettore di compact disc. Di fatto, la rappresentazione su schermo è così realistica che ci si potrebbe meravigliare di vedere apparire un file requester invece di un cassetto a slitta quando si preme sul pulsante Open del lettore CD. Il CD Player Simulator risiede in una propria finestra, che è alta circa un terzo dello schermo ed è in grado di girare in ambiente multitasking.

Compact Disc

Il QuasarSound non è mutato negli ultimi due anni, sebbene ora venga commercializzato senza hardware. QuasarSound funziona con i campionatori Perfect Sound 2 o A.M.A.S., ma è in grado di campionare un solo canale alla volta. I suoi quattro buffer corrispondo ai quattro canali audio di Amiga ed è possibile riascoltare dei suoni campionati stereo attivando due o più canali con campioni sincronizzati. QuasarSound si fonda su una interfaccia grafica semplice da usare che combina menu a discesa e pulsanti. Il suo unico motivo di gloria è la sua funzione PlayDisk. Questa consente di registrare un campione mono direttamente su floppy in un formato non standard. Quando si fa il boot dal disco, il campione viene automaticamente eseguito, Si possono collegare fra loro, a questo modo, fino a quattro dischetti, ma sono necessari quattro floppy drive per farlo. La funzione PlayDisk supporta velocità di campionamento di 19000 campioni al secondo. La limitazione sulla velocità di campionamento dipende dalla velocità con cui i suoni campionati possono essere estratti dal disco. La durata dei campioni dipende dalla velocità di campionamento. Su un singolo floppy a 10000 campioni al secondo possono stare 90 secondi di musica. Si può anche scegliere un messaggio che scorre su video quando la musica viene eseguita.

A-Sound 2.0

A-Sound 2.0 è un editor di campioni sonori monofonici da usarsi con l' hardware di Perfect Sound 2 o di A.M.A.S. La Deltaware sta lavorando a una nuova versione in grado di funzionare con Perfect Sound 3.0. Il software può gestire fino a 16 campioni alla volta. A-Sound comprende un certo numero di effetti che non sono disponibili sugli altri pacchetti. Viceversa manca del supporto diretto per gli strumenti a più ottave. La carenza più significativa di A-Sound sta nell'impossibilità di monitorare e modificare i livelli di registrazione senza dover fare un campione ed esaminare il risultato.

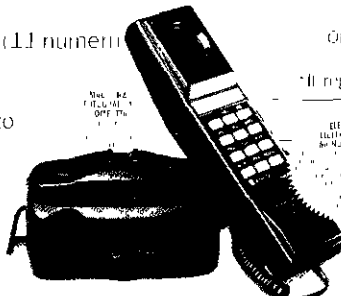
Una visita a Rio

Il campionatore fornito con lo Stereo Sample Studio della Rio Datel è contenuto in una scatola di plastica dotata di un cavo piatto che si collega alla porta parallela di Amiga. L'unità si alimenta ad un'unità esterna. Un paio di chip di conversione A/D Ferranti ZN449E gestiscono il segnale audio da elaborare. Accetta segnali stereo provenienti da altri componenti o da un microfono mediante dei jack audio standard. Esiste anche un ingresso stereo DIN a cinque poli. Il livello del segnale in ingresso deve essere impostato esternamente. La compatibilità fra il digitalizzatore della Datel e il software di altre case è inconsistente. AudioMaster III accetta i dati dal canale sinistro dell'hardware Datel quando il programma viene impostato per la compatibilità con Perfect Sound 2, e dal canale destro quando viene impostato per A.M.A.S. Al momento il digitalizzatore della Datel non viene supportato da software di altre case. La lunghezza dei campioni è limitata a due buffer di 98302 byte che possono fungere come coppia per un segnale stereo o come due buffer mono. Sebbene si possa ascoltare il segnale mediante gli altoparlanti connessi ad Amiga e impostare il livello prima di registrare, non si può monitorare il segnale mediante Amiga durante la registrazione. Il pacchetto comprende il Sample Jammer Program che funziona come un rudimentale studio di registrazione a quattro tracce. Dovrebbe servire a valutare fino a quattro campioni come strumenti musicali. Uno dei quattro canali di Amiga può essere utilizzato per il campione di batteria built-in.

Conclusioni

Perfect Sound è la scelta migliore per un digitalizzatore audio stand-alone. Il suo software ha abbastanza funzioni da rendere inutile l'acquisto di altro software. Se si desidera una maggiore potenza, AudioMaster III appare come l'editor stand-alone più avanzato tecnologicamente, sebbene Digital Studio non gli sia molto lontano. AudioMaster III oltre ad offrire una eccellente interfaccia utente, è in grado di supportare la maggior parte dei campionatori audio hardware. ▲

PERCHE' CON L'ABBONAMENTO A AMIGA MAGAZINE RICEVETE SUBITO UN REGALO

Abbonandovi oggi a **Amiga Magazine** avrete la possibilità di scegliere tra due fantastici regali: un utilissimo telefono elettronico monocorpo e una pratica macchina fotografica compatta.
Affrettatevi, l'offerta è valida sino al 31/8/1991.

sconto 20% + regalo

Amiga Costruisce un Mondo

Incontro con Joseph Conti che con il software 3-D modella il futuro.

Ben e Jean Means

Ci libriamo in volo da nord e, virando, osserviamo brillare sotto di noi dei magnifici pinnacoli gotici.
Completiamo il giro e ci dirigiamo direttamente verso il teatro in stile mediterraneo...
Amiga ci offre le ali per volare in questo paesaggio immaginario di un parco giapponese che è anni avanti nel futuro e lontano migliaia di chilometri.
Prima di sollevare una sola vanga di terra, i cauti finanziatori hanno voluto gettare uno sguardo profondo su ciò che stavano per costruire: un'isola di fantasia in una città industriosa.
Per Joseph Conti e i suoi due Amiga 2500, creare una realtà alternativa è semplicemente un altro giorno di lavoro in ufficio e il suo ufficio è uno studio con Amiga, appena fuori Los Angeles e con 50 miglia di parco nazionale dietro la porta.

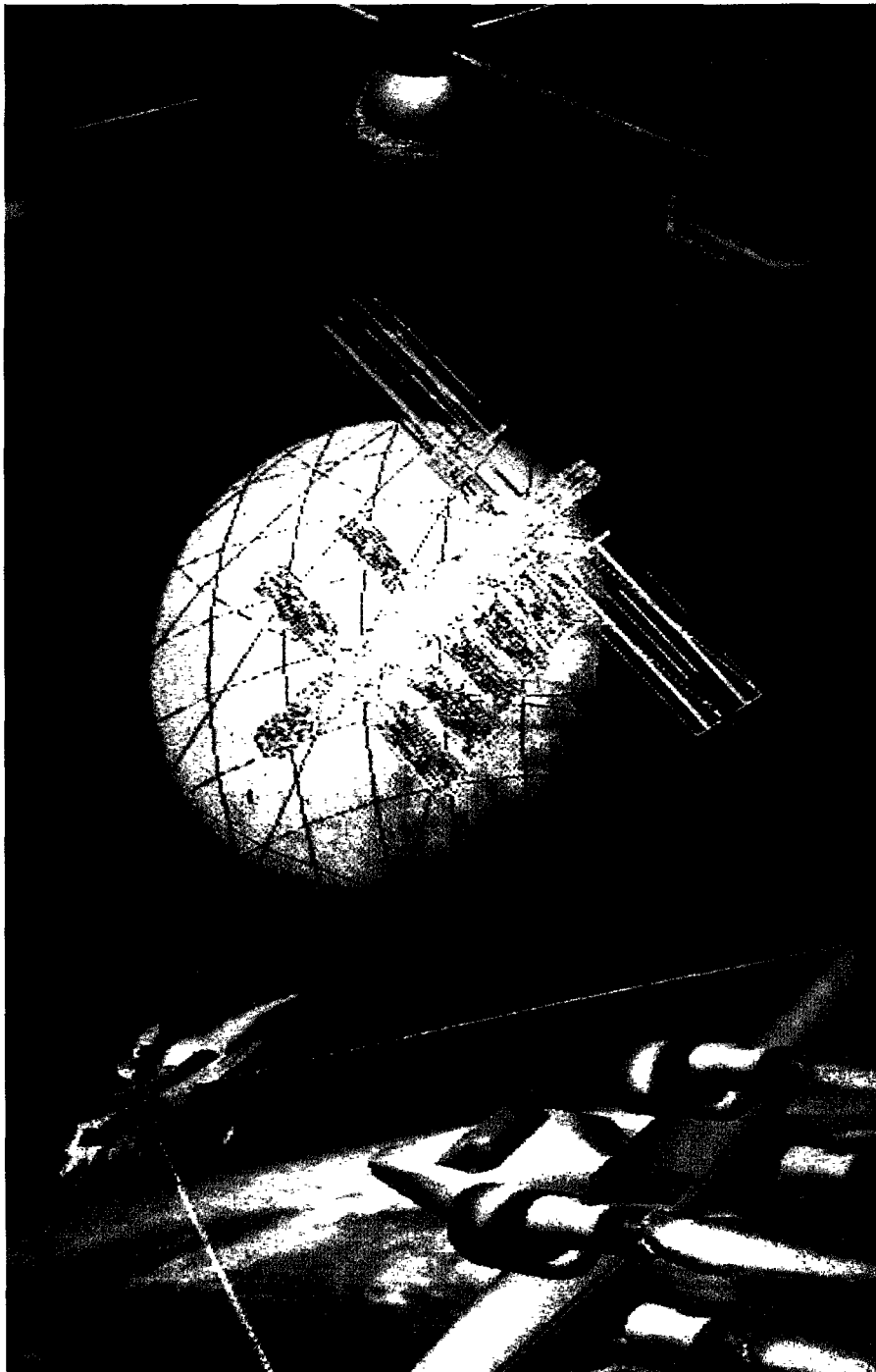
Amiga Land

Attualmente sta costruendo un parco a tema, progettato da John DeCuir Jr., il progettista dell'Epcot Center della Disney World.
"Sto modellando questo progetto da centinaia di milioni di dollari attraverso il computer" afferma Conti, guardando fuori dalla finestra, verso arbusti di rose e distese di colline in lontananza.
"Il progetto è molto simile all'Epcot Center, ma va ancora più in là, come un'intera nuova generazione di parchi a tema.
Vi compaiono molti argomenti diversi: il futuro, una combinazione di località italiane, francesi, mediterranee e di ogni parte del mondo, cosicché può dirsi universale.
E' la prima volta che Amiga viene usato in un progetto di questa scala". Pianificare un parco a tema universale, è un'impresa difficile da realizzare nel rispetto di un determinato budget.
La modellazione e l'animazione tridimensionale con Amiga abbassano questi costi pur fornendo vivide immagini del futuro.
Conti invia le cianografiche del parco a uno studio specializzato che le digitalizza e le salva come file AutoCAD IBM.
Poi carica i file AutoCAD sul suo clone IBM con 286 e le converte in file per Sculpt 4D di Amiga con il software di conversione di AutoCAD. Infine, proietta i piani bidimensionali fino all'altezza indicata sulle cianografiche e aggiunge finestre, cornicioni, guglie e parapetti.
Concluso il modello per Amiga, può elaborare visuali tridimensionali da ogni angolo o elevazione, o eseguire voli animati mozzafiato.
Conti spiega: "un modello architettonico convenzionale costerebbe circa 250000 dollari e richiederebbe probabilmente circa 9 mesi per la realizzazione.
I giapponesi amano la grafica computerizzata, più di qualsiasi altra cosa, e tale grafica avrà probabilmente il maggior impatto al momento della vendita del progetto. Ecco quant'è importante.
La grafica digitale può incidere solo del 2 per cento sull'attuale budget di progettazione, ma può contribuire al 50 per cento sulla decisione di realizzare o meno il progetto.
La parte più interessante del lavoro con Amiga è, e questo accade continuamente, che io non ho ancora in

mano una sezione modellata e ricevo una telefonata in cui mi si dice che hanno bisogno dell'immagine in Giappone nel giro di due giorni. E' così, c'è poco da discutere, e non ci sono scuse. Bisogna farlo!
Modello ciò che riesco, prendo le immagini, le sviluppo presso un servizio di sviluppo e stampa in un'ora e le invio mediante Federal Express in Giappone, il giorno successivo.
Ciò accade tutte le volte e non si potrebbe fare con qualsiasi altro computer".

Hollywood, Hollywood

Grazie al basso costo e alla facilità di uso di Amiga, Conti può mostrare una presentazione su misura ad ogni nuovo probabile cliente. Disney voleva della grafica per un nuovo pilota simile a quella che la Video Image, un leader nel settore, aveva realizzato per l'epopea spaziale "2010", si trattava di un'immagine wire-frame in cui ci si allontanava dall'astronave Discovery, passando attraverso una serie di triangoli rossi.

"La Video Image ha prodotto l'effetto originale per circa 6000 dollari e io l'ho riprodotto in dieci minuti esatti. L'ho portato alla Disney e ho detto, 2010, eccolo: l'ho fatto in dieci minuti. Se dovessi farvelo pagare vi costerebbe 600 dollari ed ecco ciò che otterreste. Suscitai il loro interesse!". Mentre la sua fama cresceva grazie ad Amiga, Joseph Conti è andato molto lontano a Hollywood.
Solo due anni fa, arrivò con un Amiga 1000 da 2.5 Mb sotto il braccio e il suo diploma in cinema dell'Università di Bridgeport, nel Connecticut, sotto l'altro. Due settimane e migliaia di telefonate più tardi, Mark Stetson della Boss Film Studios lo chiamò per gli effetti speciali di "Die Hard".
Il sogno della vita di Conti si era realizzato, e presto cominciò a scoprire come Amiga poteva migliorare le tecniche datate degli effetti speciali. Sfortunatamente, il gruppo della Boss non era pronto per le sue idee.
Nel maggio 1988 Conti lasciò la Boss per tentare di sfondare come produttore indipendente di effetti speciali e trovò un partner disponibile in John DeCuir, un production designer le cui credenziali comprendono film come "2001", "Earth Star Voyager" e "Top Gun".
"L'introdussi ad Amiga, e l'amò a prima vista, perché Amiga era in grado di gestire tutto ciò di cui aveva bisogno e di cui non riusciva a venire a capo.
Aveva bisogno di diapositive per le presentazioni, ma non poteva aggiungervi titoli in colore.
Doveva stendere i titoli sulla carta, fotografarli e otteneva ancora titoli in bianco e nero. Io "sguainai" Deluxe

Paint, creai qualche logo, e - BAM! - realizzai della grafica digitale. Quando la trasformai in una diapositiva, fu la migliore che egli aveva mai visto.
Amiga supera di gran lunga i suoi concorrenti ed è estremamente economico.
Alcuni studi a Los Angeles volevano 600 dollari per una diapositiva, e io le realizzavo a 15 dollari l'una". Conti passò dalle presentazioni con diapositive alle animazioni bidimensionali e alla grafica e animazioni 3-D. Ben presto Amiga cominciò a far di tutto per DeCuir e Conti trovò un posto sicuro come suo artista grafico su Amiga.

Strumenti odierni per i giocattoli di domani

La Mattel Toy voleva un sistema grafico digitale per realizzare i prototipi dei giocattoli e tutti, dalla WaveFront all'Alias, volevano concludere l'affare.
In questa vasta arena la Mattel scelse Joseph Conti e il suo sistema su Amiga, che immediatamente cominciò a far risparmiare all'azienda tempo e denaro.
Conti spiega: "Tradizionalmente, la Mattel noleggiava animatori per realizzare a mano tutti i disegni dell'animazione e li trasferiva su video in un secondo momento.
L'animazione implicava mesi di lavoro e costava migliaia di dollari di ora-uomo.

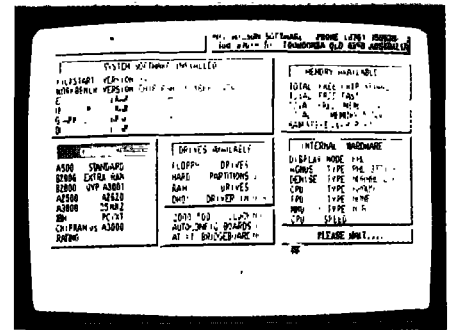
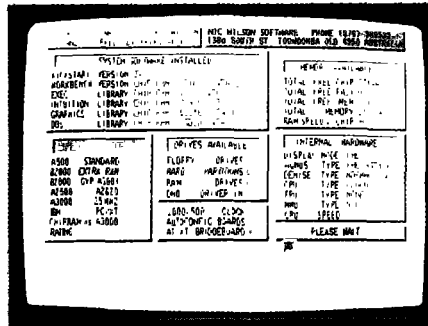
Ma l'animazione di DPaint fece avanzare la compagnia di circa 1000 anni rispetto ai suoi tempi.
La Mattel usa l'animazione nel design dei prodotti per mostrare ai bambini come funziona il gioco, o può digitalizzare un prototipo con Digi-View e poi animarlo facendolo muovere".
"I dirigenti usano nastri realizzati con Amiga per iniziare a programmare le strategie di marketing. Di fatto Amiga sta risolvendo loro molti problemi e fra questi la compressione dei tempi di lavorazione è la più importante.
Un giocattolo richiede due anni dall'ideazione alla produzione; così adesso stanno progettando i giochi per il Natale che verrà fra due anni. Hanno bisogno di comprimere al massimo questi tempi e Amiga lo sta realmente permettendo.
Ora possono fare le animazioni in pochi minuti invece di pochi mesi. E con la modellazione 3-D, possono realmente rendersi conto di come verrà il prodotto prima ancora di aver terminato il disegno".

Il vantaggio 3-D

Conti pagò 300 dollari un Amiga 1000 usato, all'inizio del 1987 ed esso gli fece guadagnare 30000 dollari il primo anno.
Conti ammette francamente: "Sono uno di quegli artisti che non possono disegnare tutto a mano libera, ma posso farlo con i mattoni da costruzione di un programma 3-D.
Posso visualizzare oggetti in 3-D, aggiungere colori e superfici e poi lasciare che il computer si preoccupi dell'ombreggiatura, dell'illuminazione e della prospettiva.
Non c'è bisogno di essere artisti per lavorare in 3-D; si deve invece essere come un regista cinematografico che se ne intende di messinscena, di movimenti e di riprese.
In 2-D devi essere un animatore che sia in grado di disegnare a mano libera ogni figura e poi muoverla. Ma in 3-D, io la costruisco, e il computer fa tutto il resto".

Il futuro

Per ora, Conti è felice del suo Amiga, ma ha forti dubbi sul modo di espanderlo.
"Amiga ha bisogno di una scheda grafica che supporti 16 milioni di colori e una risoluzione di 2000x2000.
Il solo problema è che ci deve essere uno standard. Apple possiede uno standard grafico nel suo ambiente QuickDraw.
Così tutto funziona: sia a 4, 8 o qualsiasi altro numero di bit, quello standard supporta tutto.
Se DPaint avesse una risoluzione 2000x2000 e 16 milioni di colori, tutti lo userebbero, in tutto il mondo, come il numero 1 fra i pacchetti grafici".
La Commodore risponderà a questa pressante richiesta di un grafico professionista?
Speriamo che sia così, perché, se lo facesse, noi tutti continueremmo a trarre dei benefici da quel computer che aiutò Joseph Conti a trasformare il suo sogno in oro di Hollywood.
Conti, il cui successo è stato documentato dalle pagine di "Forbes" e di "PC Weekly", condivide questa visone: "Posso affermare con sicurezza che nei prossimi anni, il 3-D diverrà il massimo strumento di design per tutti.
Verrà usato nel design industriale, nella realizzazione di prototipi per qualsiasi prodotto, dai pomelli per le porte, ai chiodi.
Gli architetti useranno il 3-D nella costruzione di case, di città, di mondi.
Sarà il primo strumento di design e le persone che giungano adesso a dominarlo saranno i designer di punta negli anni a venire". ▲

ON DISK è una rubrica mensile di quattro pagine che possono anche essere staccate e conservate, in queste pagine sono descritte tutte le informazioni dei programmi inclusi nel disco, complete di istruzioni, trucchi ecc... In questo spazio troveranno posto giochi, utility e tutto ciò che può fare Amiga.

• Games

Minimaze 2.1

Minimaze è un gioco molto semplice, in cui dovete uscire da un labirinto di 400 stanze.
Subito dopo il caricamento, tramite il solito doppio click sull'icona del programma, apparirà sullo schermo del Workbench una piccola finestra con la rappresentazione tridimensionale del labirinto e una serie di cinque gadget.
I quattro gadget con le frecce servono per muoversi, mentre il gadget rettangolare ha una duplice funzione: visualizza la direzione in cui si sta guardando (nord, sud, est e ovest) e cliccandoci sopra si ha accesso al menu delle opzioni.
Per quanto riguarda il movimento, le due frecce laterali servono per girarsi di 90 gradi a sinistra o a destra, la freccia verso il basso serve per girarsi di 180 gradi, mentre la freccia verso l'alto serve ad avanzare nella direzione in cui si sta gurdando.
Il menu delle opzioni è composto di 4 gadget: Quit, Load Maze, Save Maze e Help. Servono, rispettivamente, per uscire dal programma, per caricare un labirinto (sul disco è presente un labirinto di esempio con il nome sample.maze), per salvare una situazione di gioco e per visualizzare la mappa dell'intero labirinto.

MazeMaker 2.0

Con il programma Maze Maker si possono realizzare dei labirinti personalizzati o modificare quello sul disco. L'uso di questo programma è molto semplice e tutti i tasti da utilizzare sono visualizzati sullo schermo. Non essendoci spazio sul disco di Amiga Magazine è necessario salvare i propri labirinti su un altro disco.

SPECIFICHE DEL PROGRAMMA

CONFIGURAZIONE MINIMA
512K RAM
Kickstart 1.2/1.3

UTILIZZO
Workbench: Doppio click sull'icona
CLI/SHELL: CD Amigamagazine:
games/maze
RUN Minimaze2.1
RUN Mazemaker2.0

FILE DI SUPPORTO
Arp.library

Ballbench

BallBench è un gioco che sta a metà strada fra Arkanoid e il ping pong.
Il caricamento si attua come al solito con il doppio click sull'icona del programma. Dopo alcuni istanti appaiono sullo schermo del Workbench due volti umani: a sinistra una donna e a destra un uomo.
Cliccando sul tasto destro del mouse, abbiamo accesso ai menu.
Nel menu Project abbiamo l'opzione About che visualizza un breve messaggio sugli autori del programma e l'opzione New Game per iniziare una nuova partita. Tramite il menu Mode selezioniamo il tipo di gioco: Direct, Ping Pong e Bounce. Il primo è il modo di default, con la palla che segue traiettorie rettilinee, rimbalzando solo sui bordi dello schermo. Il secondo simula il movimento della palla su un tavolo da ping pong.
Il terzo simula la presenza della forza di gravità, attirando la palla verso la parte bassa dello schermo.
Tramite il menu Control selezioniamo il metodo di controllo dei due giocatori (joystick, mouse, tastiera o computer) e il livello di gioco, da facile a impossibile. Per terminare basta cliccare nel gadget di chiusura della finestra con i punteggi.

SPECIFICHE DEL PROGRAMMA

CONFIGURAZIONE MINIMA
512K RAM
Kickstart 1.2/1.3

UTILIZZO
Workbench: Doppio click sull'icona
CLI/SHELL:CD Amigamagazine:
games/ballbench
RUN Ballbench

FILE DI SUPPORTO
Tutti nella directory Ballbench

Spheroids

Nell'anno 2100, dopo la Terza Guerra Mondiale, i superstiti dello scontro nucleare si affrontano in duelli in cui ogni paese è rappresentato da un gladiatore.
Luogo delle dispute è l'Arena, un vasto campo di battaglia con dossi e colline.
All'inizio di ogni duello, sei bombe vengono disposte casualmente nell'Arena.
Ogni gladiatore, muovendosi col proprio veicolo, deve prendere le bombe e scagliarle contro l'avversario.
Le bombe si muovono velocemente ma la superficie irregolare del terreno rende imprevedibile la loro traiettoria. L'unico modo di uscire vittoriosi è avere riflessi veloci e molta fortuna.
Per caricare Spheroids fate un doppio click sulla sua icona. Subito dopo il caricamento, apparirà sullo schermo il campo di battaglia e l'elenco dei tasti che controllano le opzioni del gioco.
Inserite uno o due joystick e sceglite il campo di battaglia su cui confrontarvi tramite i tasti funzione da 1 a 7.
La seconda scelta riguarda il tipo di gioco: F8 per vedere una demo in cui il computer gioca con se stesso, F9 per giocare con il computer, F10 per giocare con un vostro amico. La battaglia ha subito inizio, come sul ring del pugilato, ai due angoli ci sono i contendenti: in alto a destra il veicolo verde controllato dal giocatore 1 e in basso a sinistra il veicolo blu controllato dal giocatore 2. Nella parte alta del video sono indicate le vite rimaste.
Casualmente sullo schermo verrà disposto un teletrasporto che servirà a toglervi dalle situazioni più difficili, ma fate attenzione perché anche il movimento delle bombe è influenzato dalla sua azione. Per utilizzare le bombe è sufficiente passarci sopra per collezionarle (una alla volta); per lanciarle contro il nemico basta premere il tasto di fuoco tenendo il joystick nella direzione voluta. La bomba assumerà il colore del veicolo che l'ha lanciata e solo quando tornerà di colore rosso sarà riutilizzabile. Alla fine del gioco il vincitore sarà indicato dalla barra lampeggiante, nella parte alta del video.
In ogni momento potete premere il tasto Esc per uscire dal gioco.

SPECIFICHE DEL PROGRAMMA

CONFIGURAZIONE MINIMA
512K RAM
Kickstart 1.2/1.3

UTILIZZO
Workbench: Doppio click sull'icona
CLI/SHELL:CD Amigamagazine:
games/spheroids
RUN Spheroids

FILE DI SUPPORTO
Tutti nella directory Spheroids

• Utility

AIBB2

AIBB2 è un programma che serve a testare le performance della CPU del vostro Amiga. Esso non tiene conto dell'eventuale presenza di coprocessori matematici. Il programma è molto semplice da usare dato che le opzioni vengono selezionate da menu, mentre i test vengono fatti partire cliccando sui gadget nella parte bassa dello schermo.
I sei test eseguibili sono:
-WritePixel: disegna un box sullo schermo un punto alla volta e lo cancella nello stesso modo.
-Sieve: è il test di Eratostene per i numeri primi.
-Sort: ordina 12000 numeri interi con un algoritmo shell-sort.
-Savage: esegue un certo numero di operazioni su dei numeri in virgola mobile.
-Dhrystone: esegue il test Dhrystone comune alla maggior parte dei computer.
-Matrix: esegue un test di addizione e moltiplicazione su tre matrici di 40x40 numeri interi.
Questi test danno una idea generale delle performance di Amiga.
Molto spesso i risultati sono sfalsati da particolari condizioni del sistema operativo. Per questo motivo si raccomanda di non far girare programmi in background mentre si eseguono i test. Consigliamo anche di selezionare l'opzione 'Be Selfish' nei menu e di liberare il maggior quantitativo di FAST RAM prima di far girare il programma.

I Dispositivi di Input

Notizie essenziali dal bunker dei Metal Basher per Amiga (parte III)

Andrew Warlond

Ancora sulla breccia, metal basher! Questa volta, penso che sia giunto il momento di discutere il metodo sbarazzino usato dai metal basher per ricevere input dal mondo esterno. I dispositivi che intendiamo utilizzare sono la tastiera, il joystick e il mouse. Anche il disco è un dispositivo di input, ma è piuttosto complicato, così ne discuteremo separatamente in un prossimo articolo.

Dapprima, la tastiera. E' una "bestia" piuttosto complicata, ma le routine per l'accesso diretto, una volta scritte, sono brevi e pulite. La tastiera di Amiga è connessa a un chip speciale, il CIA (Complex Interface Adapter) 8520. Ve ne sono due attualmente, chiamati CIAA e CIAB. Sono chip molto complessi e utili, impiegati dal sistema per molti scopi. Una descrizione completa di tali chip va oltre i limiti di questo articolo, così prenderemo in considerazione solo i registri CIAA di cui abbiamo bisogno per accedere alla tastiera.

Prima di discutere i dettagli, sarà meglio fare un breve riassunto di tutto quel che capita quando si schiaccia un tasto. Quando un tasto viene premuto o rilasciato, un codice speciale a 8 bit viene inviato mediante una linea seriale, un bit dopo l'altro, ad un particolare registro CIAA. Questo chip genera poi un interrupt di livello 2 per il 68000, che lo avvisa del fatto che è terminata la ricezione del codice.

La routine di gestione dell'interrupt deve prendere questo codice dal registro e salvarlo, poi deve inviare uno speciale segnale di handshake alla tastiera per informarla che CIAA è pronto a ricevere il prossimo codice.
Abbiamo bisogno dei seguenti registri CIAA:

Nome	Descrizione	Indirizzo
SDR	Serial Data Register	BFEC01
TALO	Timer A LOw byte	BFE401
TAHI	Timer A HIgh byte	BFE501
CRA	Control Register A	BFEE01
ICR	Interrupt Control Register	BFED01

Il codice inviato dalla tastiera giunge nel Serial Data Register. Il Timer A è uno dei due timer (l'altro è il Timer B) presenti in CIA e noi lo useremo sia per sincronizzarci con l'arrivo dei dati in SDR, sia per calcolare la durata del segnale di handshake da restituire alla tastiera. Il Control Register A viene usato per configurare il timer e il Serial Data Register. L'Interrupt Control Register è un registro a lettura e scrittura. Scriveremo in esso per indicare a CIA le condizioni che devono generare un interrupt di livello 2 per il 68000, il quale, a sua volta, porrà in esecuzione la nostra routine. Lo leggeremo per determinare le cause effettive dell'interrupt.

Ora prendiamo in considerazione il tipico ciclo di ricevimento di un codice dalla tastiera e di invio del segnale di handshake. La figura 1 mostra ICR in maggior dettaglio. Uno degli interrupt consentiti è quello che viene generato quando SDR è pieno. Questo avviene quando è stato ricevuto da CIA un codice di tastiera completo. CIA emette allora un interrupt di livello 2 per il 68000, che farà partire a sua volta l'esecuzione della nostra routine.

Il primo compito della nostra routine è leggere ICR per determinare che cosa abbia generato l'interrupt. In questo caso, il bit SP alto indica che l'SDR è pieno. Poi, la nostra routine legge il codice in SDR e lo ripone in un buffer che dovrà essere usato dal programma principale.

Dobbiamo ora dire alla tastiera che abbiamo ricevuto il

Interrupt Control Register (ICR)

Bit	Nome	Descrizione
7	S/C	In scrittura: 0 = pone a zero i bit richiesti 1 = pone a uno i bit richiesti
6	X	Non utilizzato
5	X	Non utilizzato
4	FLAG	Interrupt flag
3	SP	Interrupt porta seriale
2	ALRM	Interrupt Alarm
1	TB	Interrupt Timer B
0	TA	Interrupt Timer A

Fig. 1

Control Register A

Bit	Nome	Descrizione
7	X	Non utilizzato
6	SPMODE	Modo porta seriale: 0 = Input - 1 = Output
5	INMODE	Seleziona il tipo di segnale che deve considerare il Timer A 0 = impulsi "02" 1 = transizioni +VE
4	LOAD	Impone il caricamento del Timer A
3	RUNMODE	Modo Timer A: 0 = continuo - 1 = non continuo
2	OUTMODE	Modo Output: 0= impulsi
1	PBON	Output Timer: 0 = operazioni normali - 1 = PB6
0	START	Controllo Timer A: 0 = Fine - 1 = Inizio

Fig. 2

Registri JOYODAT e JOY1FDAT

Bit	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Nome	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	X7	X6	X5	X4	X3	X2	X1	X0

Fig. 3

byte, inviandole un segnale di handshake. Per far questo dobbiamo modificare il modo di SDR in OUTPUT, impostando il bit SPMODE in CRA. La figura 2 mostra CRA in maggior dettaglio. SDR deve essere lasciato in modo OUTPUT per almeno 85 microsecondi, perché il segnale di handshake venga riconosciuto dalla tastiera. Pertanto usiamo il Timer A per calcolare questo periodo.

Programmiamo TA perché conti gli impulsi "02", impostando il bit INMODE in CRA. Gli impulsi "02" vengono generati dall'hardware di Amiga e viaggiano a circa 1/10 della velocità del processore. Questo dà approssimativamente 1.397 microsecondi per impulso. Così, per un ritardo di 85 secondi, dobbiamo immettere un valore iniziale di almeno:

85 / 1.397 = 61

nel Timer A. Il Timer A viene impostato nel modo ONESHOT mediante CRA, ciò significa che esso conterà fino a zero una sola volta, poi si fermerà. Facciamo anche generare a CIA un altro interrupt quando TA sia a zero, impostando il bit TA in ICR. A questo punto possiamo tornare al programma principale finché CIA non ci interromperà di nuovo.

In questo secondo interrupt, controlleremo se la sua causa è stata il bit TA di ICR. In tal caso sappiamo che l'handshake è stato inviato correttamente e possiamo preparare CIA a ricevere un altro codice dalla tastiera. Tutto quello che dobbiamo fare è cambiare SDR in modo INPUT e abilitare di nuovo l'interrupt SP in ICR. Dopodiché, possiamo tornare al programma principale finché non verrà completata la ricezione di un altro codice dalla tastiera. Phew!!!! Vi avevo avvertiti che era complicato! Ma se si studia il codice sorgente che si trova su disco, rileggendo il discorso appena concluso, penso che tutto diventerà chiaro.

Si noti che i codici inviati dalla tastiera non sono ASCII. Essi forniscono solo informazioni sulla posizione del tasto che è stato premuto. Il carattere stampato su quel tasto può variare da paese a paese e anche fra i diversi modelli di Amiga. L'Hardware Reference Manual contiene una lista di codici per i modelli A500/1000/2000 che sarebbe troppo lungo per essere presentato in questa sede. Se avete bisogno solo di qualche tasto fate degli esperimenti con le routine d'esempio, per vedere che cosa viene restituito dalla pressione dei diversi tasti.

Passiamo ora alla discussione del joystick e del mouse. Siccome usano la stessa interfaccia, vi si accede in maniera simile.

La figura 3 mostra il formato e gli indirizzi di due registri hardware, JOY0DAT e JOY1DAT. Questi permettono di accedere a contatori a 8 bit per i valori X e Y delle porte joystick 0 e 1. Quando il mouse viene mosso, questi contatori vengono incrementati o decrementati a seconda della direzione del movimento.

Si noti che essi ricominciano da capo quando raggiungono il massimo o il minimo del valore consentito e di ciò bisognerà tener conto nel codice. Il modo migliore per gestirli è salvare le coordinate assolute del mouse rispetto allo schermo e il precedente valore dei contatori. Quando questi contatori vengono ricontrollati si dovranno semplicemente sottrarre i valori precedenti da quelli nuovi e aggiungere il risultato alle coordinate assolute dello schermo. Questo funziona se il tempo tra due successivi campionamenti dei registri non è troppo elevato. Per questa ragione, la routine esemplificativa presente su disco dovrebbe essere usata come parte della routine di gestione dell'interrupt di vertical blank.

Se alla porta è connesso un joystick, si può stabilire la sua posizione effettuando un po' di calcoli logici sui registri JOYxDAT:

Per determinare	Operazione (si veda la figura 3)
Avanti	Y1 xor Y0
Indietro	X1 xor X0

(segue a pag. 50)

Un primo approccio al misterioso mondo degli hunk

Parte prima: nella terra del loader

Romano Tenca

Hunk e multitasking

I file utilizzabili con Amiga devono, in certi casi, attenersi alle specifiche decise dalla Commodore e implementate mediante l'AmigaDOS; queste regole dettano un formato particolare per due soli tipi di file: i cosiddetti file oggetto, quelli che terminano di solito con ".o", e i file eseguibili. Tutti gli altri file, come immagini, testi, suoni, dati grezzi, non vengono riconosciuti dall'AmigaDOS e se anche rispettano uno standard, come l'IFF, è responsabilità del programma utente riconoscerli e interpretarli. I file eseguibili, invece, devono avere una particolare struttura affinché il loader del DOS, che si attiva con la funzione LoadSeg( ) dell'AmigaDOS, possa caricarli in memoria e renderli adatti all'esecuzione.

La struttura dei file oggetto è invece necessaria per il linker ufficiale dell'AmigaDOS (ALINK), che ha il compito di trasformare uno o più file generati da uno o più compilatori (di qualsiasi linguaggio) in un unico file eseguibile:

sorgente -> compilatore -> file oggetto -> linker ->
file eseguibile -> loader -> programma eseguibile in memoria

Il motivo fondamentale, ma non il solo, per cui i file eseguibili devono avere una particolare struttura interna, è il sistema multitasking. Infatti, a causa sua, Amiga deve poter caricare un programma a qualsiasi indirizzo in memoria: non si può sapere in anticipo dove sarà la memoria libera quando il programma verrà lanciato, perché in qualsiasi punto della memoria ci può essere un programma che gira in multitasking; né il compilatore, né il linker, dunque, possono sapere quale sarà l'area libera di memoria.

Una soluzione a questo problema può essere raggiunta usando, per il codice, esclusivamente riferimenti a 16 bit relativi al PC; ciò, ovviamente, limiterebbe enormemente le possibilità dei compilatori e dei programmatori e impedirebbe l'indirizzamento diretto di dati posti a una distanza superiore a +/-32K. Tale opzione viene usata qualche volta da compilatori assembler per piccoli programmi, i quali risultano, dunque, completamente indipendenti dalla posizione in memoria (non sono cioè file rilocabili, ma ben di più).

Se si vogliono usare degli indirizzi di memoria a 32 bit, bisogna seguire un'altra strada, che consiste nel salvare, assieme al codice, delle informazioni che permettano di aggiustare i riferimenti a 32 bit a seconda della posizione del programma in memoria. Per capire come funziona il meccanismo, facciamo un esempio in assembler:

```
$0000           JMP PROVA
 ....
$0100 PROVA     MOVE D0,D1
```

In questo caso, all'offset $0000 del programma viene effettuato un salto alla routine PROVA posta all'offset $0100. Il salto utilizza un indirizzo di memoria assoluto, dunque a 32 bit, nel senso che l'istruzione JMP viene seguita da una longword che contiene l'indirizzo PROVA. Ora, come abbiamo già detto, né il linker, né il compilatore possono sapere in che posizione in memoria sarà situato il programma e, di conseguenza, l'indirizzo PROVA. Per risolvere il problema Amiga memorizza nel codice, dopo l'istruzione JMP, l'offset di PROVA rispetto all'inizio del programma, in questo caso $00000100, poi aggiunge al file delle informazioni che saranno usate dal loader per aggiustare il riferimento, dopo aver caricato il file in memoria.

Infatti, in quel momento basterà aggiungere al valore posto dopo l'istruzione JMP (cioè $00000100) l'indirizzo iniziale del codice: se, ad esempio, il codice venisse allocato all'indirizzo $00C00000, il loader sommerà tale valore a $00000100 ottenendo l'indirizzo $00C00100 e il programma assumerà questo aspetto:

```
$00C00000        JMP $00C00100
 ....
$00C00100        MOVE D0,D1
```

in cui il riferimento appare conforme alle nostre aspettative, qualsiasi sia l'indirizzo di memoria in cui sia stato effettivamente caricato il programma.
E' chiaro allora che il file eseguibile non potrà contenere solo il codice, ma anche delle informazioni di rilocazione, per cui, nel nostro esempio, il file sarà composto da almeno due unità autonome e distinte, una con il codice, l'altra con le informazioni di rilocazione. Questi "pezzi" di file sono detti "hunk".

Gli hunk non servono solo a rilocare il codice internamente, vengono usati comunemente anche per aggiustare i riferimenti fra parti di programma che risiedono in zone non contigue di memoria. Infatti, come si sa, Amiga può caricare in memoria un programma diviso in pezzi (hunk, appunto) posti in locazioni diverse di memoria, questo per evitare di dover allocare grosse regioni contigue di RAM; quest'ultime, infatti, possono non essere presenti nel sistema, anche quando la RAM disponibile è sufficiente, sempre a causa del multitasking e della sua incessante opera di allocazione e rilascio di aree di memoria.

Ma gli hunk servono a gestire anche altri tipi di informazioni, come quelle necessarie al linker per risolvere i riferimenti a simboli esterni, quelle per il debugger, quelle per gli overlay.

Il formato degli hunk è stato reso pubblico nel 1987 dalla Commodore nel libro "The AmigaDOS Manual" della Banthams Book, tradotto in italiano dalla IHT. Non è stato più aggiornato da quella prima edizione.

In generale, gli hunk sono costituiti da una serie di longword: anche le stringhe vengono sempre (almeno negli hunk standard) allineate alla longword e portate alle dimensioni necessarie con una serie eventuale di zeri finali.

La prima longword di ogni hunk ne identifica il tipo: per conoscerne il valore, che corrisponde al nome simbolico che useremo nel testo, si faccia riferimento alla figura 1, oppure si usi il programma hunk.rexx scritto in ARexx che accompagna, su disco, quest'articolo.

Figura 1: hunk noti e meno noti

dec	hex	nome simbolico	
998	3E6	hunk_resident	
999	3E7	hunk_unit	
1000	3E8	hunk_name	
1001	3E9	hunk_code	
1002	3EA	hunk_data	
1003	3EB	hunk_bss	
1004	3EC	hunk_reloc32	
1005	3ED	hunk_reloc16	
1006	3EE	hunk_reloc8	
1007	3EF	hunk_ext	
1008	3F0	hunk_symbol	
1009	3F1	hunk_debug	
1010	3F2	hunk_end	
1011	3F3	hunk_header	
1013	3F5	hunk_overlay	
1014	3F6	hunk_break	
1016	3F8	hunk_extLatt	(Lattice)
1018	3FA	hunk_libnode	(Lattice)
1019	3FB	hunk_libname	(Lattice)

I file eseguibili

I file eseguibili sono costituiti da un insieme di hunk di diverso tipo. Fra gli hunk che compongono un file eseguibile è possibile stabilire una specie di gerarchia, nel senso che certi hunk possono comparire solo in dipendenza di altri e che alcuni sono obbligatori, altri facoltativi.

Quali siano queste dipendenze emergerà, man mano, dall'analisi che effettueremo dei singoli hunk. Ma prima di cominciare, vale la pena di precisare che i file eseguibili sono sostanzialmente di due tipi: quelli normali (diciamo così) e quelli ad overlay.

Non possiamo soffermarci a spiegare in maniera analitica il funzionamento di un file ad overlay, diciamo solamente che si tratta di un file organizzato in modo tale che, nel corso dell'esecuzione, il codice da eseguire venga tratto dal file solo nel momento in cui ce ne sia effettivamente bisogno, per poi essere rilasciato al termine dell'utilizzo. Il sistema viene usato solo da programmi molto lunghi, per limitare l'occupazione di memoria.

hunk_header

Vediamo ora l'effettiva struttura di un file eseguibile e, per seguire meglio il discorso, prendiamo in considerazione il file BINDDRIVERS dell'AmigaDOS, ecco le sue prime longword:

```
00000  000003F3    hunk_header (nodo primario)
00001  00000000    marcatore
00002  0000000B    numero totale di hunk del file
00003  00000000    primo hunk di questo nodo
00004  0000000A    ultimo hunk di questo nodo
00005  00000076    memoria per hunk n. 0
00006  00000004    memoria per hunk n. 1
00007  000000FF    memoria per hunk n. 2
00008  00000015    memoria per hunk n. 3
00009  00000023    memoria per hunk n. 4
0000A  00000000    memoria per hunk n. 5
0000B  00000029    memoria per hunk n. 6
0000C  0000001D    memoria per hunk n. 7
0000D  00000010    memoria per hunk n. 8
0000E  00000015    memoria per hunk n. 9
0000F  00000000    memoria per hunk n.10
```

La prima longword è l'identificatore hunk_header; l'header identifica un nodo, e il primo nodo è detto nodo primario; in un file normale esiste solo il nodo primario e questo hunk compare una sola volta all'inizio del file, mentre in un file ad overlay compaiono più nodi e dunque più hunk_header.

Questo hunk è composto da una serie opzionale di nomi di librerie residenti che terminano con una longword a 0; siccome queste librerie non vengono solitamente usate nei file eseguibili, eviteremo di approfondire l'argomento (chi

fosse interessato può consultare l'articolo apparso nelle pagine di Transactor in Amiga Magazine n. 13, 1990). L'identificatore dell'hunk è quindi seguito, di solito, direttamente da uno 0, come avviene in questo caso.

La longword successiva indica il numero totale di hunk del file che il loader dovrà caricare in memoria, in questo caso 11 ($0B). Questo numero si riferisce agli hunk del tipo hunk_code, hunk_data e hunk_bss (oltre che agli hunk delle eventuali librerie residenti), i soli per i quali va allocato dello spazio in memoria, cioè non comprende gli altri hunk che contengono informazioni di rilocazione, debugging e così via.

La longword successiva indica il numero progressivo del primo hunk da caricare (0) e quella seguente il numero dell'ultimo hunk ($0A) di questo specifico nodo; per sapere dunque da quanti hunk sia composto questo nodo si dovrà sottrarre 0 a 10 e aggiungere 1 (il risultato sarà 11, in questo caso). E' importante notare che tale valore può non corrispondere al numero totale di hunk da caricare in memoria indicato dall'header: questo accade quando esistono delle librerie residenti o, più spesso, quando si è di fronte ad un file con struttura ad overlay. Anche per quest'ultimo caso vale la pena di fare di fare un esempio, questa volta immaginario:

```
00000  000003F3  hunk_header (nodo primario)
00001  00000000
00002  00000006  numero totale di hunk del file
00003  00000000  primo hunk di questo nodo
00004  00000001  ultimo hunk di questo nodo
.....
00100  000003F3  hunk_header (nodo secondario)
00101  00000000
00102  00000002
00103  00000002  primo hunk di questo nodo
00104  00000002  ultimo hunk di questo nodo
.....
```

In questo file ad overlay, come vedete, compaiono due hunk_header: nel primo si segnala la presenza di 6 hunk in totale, ma il primo nodo contiene solo le informazioni relative ai primi due hunk, il numero 0 e il numero 1, quelle relative all'hunk numero 2 si trovano nel nodo seguente (alla longword $100), dove appunto si dice che il primo hunk da caricare è quello numero 2 e l'ultimo è sempre il 2 (il totale di hunk di caricare per questo nodo è 1, infatti 2-2+1=1).

E' chiaro che per arrivare al totale di 6 hunk dovranno esistere nel file altri nodi con le informazioni relative agli ultimi 3 hunk mancanti. Per segnalare al loader che il file è in overlay, esistono, comunque, due hunk specifichi che citeremo oltre.

Torniamo ora al file BINDDRIVERS: le ultime undici longword indicano rispettivamente la quantità di memoria da allocare per gli hunk che vanno dal numero 0 al numero 10, espressa in longword. E' importante sottolineare che il loader userà questi valori per allocare la memoria per tali hunk, indipendentemente dalla effettiva lunghezza degli hunk che vi dovranno poi essere copiati.

Ciò significa che è possibile che la memoria allocata ecceda l'effettiva lunghezza di un hunk. Tale "trucco" viene usato da molti programmi, ma soprattutto da certi file "crunchati", nei quali la quantità di memoria da allocare è pari alla lunghezza dell'hunk decompattato, mentre l'hunk vero e proprio è più breve, risultando, appunto, "crunchato".

A questo modo, il programma che decompatta il file trova la memoria da utilizzare già allocata dal loader del DOS. Capita anche di trovare degli slot con il valore 0, in tal caso non viene allocata memoria e l'hunk che vi dovrà essere caricato dovrà avere lunghezza 0: è come se non esistesse (ad esempio, nella icon.library, le librerie shared sono file eseguibili, ci sono 4 hunk di lunghezza 0, penso che si possano eliminare, senza danni, ma non ho mai provato).

Ma queste longword hanno un'altra peculiarità: se il bit 30 è alto, il loader assume che la memoria richiesta sia di tipo CHIP e tenta di allocare questo tipo di memoria per il programma. Se invece è alto il bit 31 la memoria da caricare deve essere di tipo FAST. I vari programmi esistenti, tipo SCATOM, HUNKWIZARD e altri ancora, che cambiano il tipo di memoria degli programmi eseguibili, non fanno nient'altro che questo: alzare o abbassare i bit citati per determinare il tipo di memoria da utilizzare (CHIP o FAST o di qualsiasi tipo) per un determinato hunk.

Si noti che questa caratteristica è stata introdotta con la versione 1.1 dell'OS. Le versioni precedenti non funzioneranno con i file di questo tipo: riporteranno l'errore 103, memoria insufficiente.

Hunk_code

Andiamo ora a vedere ora come continua il file BINDDRIVERS:

```
00010  000003E9  hunk_code
00011  00000076  lunghezza in longword
00012  48E78080  prima longword del codice
```

Il valore 3E9 segnala l'inizio di un hunk_code che contiene il codice vero e proprio. La longword successiva indica la lunghezza (sempre in longword) del codice, in questo caso $76, pari a 118 longword cioè 472 byte. I 472 byte successivi saranno, dunque, occupati dal codice che il loader dovrà copiare nella memoria riservata all'hunk 0.

E' ovvio che in un file eseguibile deve comparire almeno un hunk di questo tipo. In BINDDRIVERS, ad esempio, ne compaiono 8.
Il codice del file è cioè diviso in 8 parti ognuna delle quali

verrà caricata in una propria area di memoria. E' chiaro che se esistono, come è ovvio, dei riferimenti fra il codice di un hunk_code e quello di un altro, il sistema dovrà avere delle informazioni per aggiustare i riferimenti, come avevamo precisato all'inizio.

hunk_reloc32

Ecco cosa segue in BINDDRIVERS, $76 longword più avanti:

```
00088 000003EC hunk_reloc32
00089 00000002 2 rilocazioni da effettuare
               nell'hunk 0
0008A 00000001 rispetto all'hunk 1
0008B 00000074 primo offset in hunk 0
0008C 0000000C secondo offset in hunk 0
0008D 00000001 1 rilocazione da effettuare
               nell'hunk 0
0008E 00000002 rispetto all'hunk 2
0008F 0000016A primo offset in hunk 0
00090 00000000 fine hunk_reloc
```

Siamo di fronte a un hunk_reloc32 che serve al loader per risolvere i riferimenti a 32 bit all'interno dell'hunk o fra hunk diversi. La prima longword identifica l'hunk. Va notato che questo hunk si riferisce implicitamente all'hunk_code precedente (il numero 0) e dipende da esso. Questo hunk non richiede che il loader allochi della memoria per caricarlo e quindi non occupa uno slot di memoria nell'hunk_header iniziale: in altre parole non è uno di quegli 11 hunk cui si fa riferimento nell'header.

La longword successiva contiene il valore 2, esso indica il numero di rilocazioni a 32 bit da effettuare nell'hunk 0, ma rispetto a quale hunk? Ce lo dice la longword seguente: rispetto all'hunk 1. Dopodiché, seguono 2 offset relativi all'hunk 0, che sono gli indirizzi da rilocare.

Per capire come si comporta il loader, facciamo un esempio pratico: ammettiamo che l'hunk 0 sia stato caricato in memoria a partire dalla locazione $4000 e che l'hunk 1 sia stato invece caricato alla locazione $8000; ora il loader deve rilocare l'hunk 0 rispetto all'hunk 1: somma $4000 al primo offset ($74) e ottiene $4074: è l'indirizzo da rilocare; che valore dovrà scrivervi? Alla locazione $4074 sta un secondo offset (generato dal linker) che si riferisce all'hunk 1, poniamo $0A; ora basta sommare questo offset all'indirizzo di memoria dell'hunk 1 ($8000+$0A=$800A) e riporre il risultato nell'indirizzo $4074. La prima rilocazione è stata effettuata. Si noti che la rilocazione a 32 bit può avvenire all'interno di uno stesso hunk, per esempio fra l'hunk 0 e l'hunk 0.

La longword successiva contiene il secondo offset da rilocare. Poi segue il valore 1; se fosse stato 0 le rilocazioni sarebbero terminate, invece è 1 e ciò significa che esiste un'altra rilocazione rispetto ad un altro hunk: è l'hunk 2, come recita la longword seguente. Saltiamo l'offset $16A e giungiamo alla longword 0, che indica la fine dell'hunk_reloc.

Si noti per finire che la struttura di questo hunk è identica a quella degli hunk_reloc16 e hunk_reloc8 che compaiono (insieme all'hunk_reloc32) nei file oggetto e di cui parleremo in seguito.

hunk_end

Andiamo avanti e troviamo questa longword:

```
00091  000003F2
```

è l'hunk_end, una longword in tutto, che segnala la fine dell'hunk 0. L'hunk_end viene utilizzato per determinare la fine delle informazioni associate ad un hunk del tipo code, data o bss.
Capita di non trovarlo in certi file, quando non ci sono altri hunk associati all'hunk_code (o data, o bss).

hunk_data

La longword successiva segnala l'inizio di un hunk_data:

```
00092  000003EA  hunk_data
00093  00000004  lunghezza in longword
00094  00000000  prima longword di dati
00095  00000000
00096  00220008
00097  52580000  quarta e ultima longword di dati
```

L'hunk_data contiene i dati inizializzati del programma; non tutti, si intende: a seconda dei compilatori e dei linguaggi, certi dati possono essere riposti direttamente nell'hunk_code. In C, generalmente, i dati inizializzati sono posti in questo hunk, mentre quelli non inizializzati stanno in un hunk del tipo bss. Per l'hunk_data viene allocata della memoria dal loader, per cui il nostro contatore di hunk deve essere aggiornato, siamo ormai giunti all'hunk 1 (che è il secondo, il primo era il numero 0).

Il formato prevede che dopo l'identificatore appaia la lunghezza in longword dei dati e poi i dati stessi: è come hunk_code, da questo punto di vista. Gli hunk data possono essere rilocati, e quindi essere seguiti da un hunk_reloc32, che, comunque, in BINDDRIVERS non c'è. Il senso dell'hunk_reloc32 è sempre lo stesso e il funzionamento pure, per cui valgono esattamente le cose dette in precedenza. Anche questo hunk viene normalmente concluso con un hunk_end.

hunk_bss

Il terzo tipo di hunk per cui il loader alloca memoria è l'hunk bss. Il programma BINDDRIVERS non lo contiene, per cui facciamo un esempio immaginario:

```
001D5   000003EB   hunk_bss
001D6   0000000A   quantità di memoria da allocare
```

Dopo l'identificatore appare una sola longword, essa indica il numero di longword di memoria ($A, in questo caso) che il sistema deve allocare e azzerare per questo hunk. Praticamente, l'hunk_bss serve a comandare al loader di riservare una regione di memoria per il programma. Il suo scopo è quello di evitare di inserire nel file byte non inizializzati, che l'allungherebbero inutilmente. Le variabili esterne del C, non inizializzate, possono essere poste in questo tipo di hunk. La lunghezza può essere 0, ma la cosa viene utilizzata solo dai programmi scritti in FORTRAN per i blocchi COMMON. Negli altri programmi non è mai a 0. Questo hunk non può essere rilocato ed è normalmente seguito dall'hunk_end.

hunk_debug

In un file eseguibile possono comparire uno o più hunk_debug destinati a contenere informazioni per i debugger. Il suo formato è semplicissimo:

```
00092   000003F1   hunk_debug
00093   00000004   lunghezza in longword
00094   00000000   prima longword di dati
.....
```

Dopo l'identificatore compare la lunghezza in longword dei dati e, quindi, i dati veri e propri. L'AmigaDOS non pone alcuna ulteriore restrizione sul loro uso. Ovviamente, il loader non carica in memoria i dati di questo hunk, sta al debugger operare su di loro. E' evidente che tale hunk si presta anche ad usi diversi: per esempio, un cruncher potrebbe inserirvi delle informazioni utili al momento della decompattazione.

hunk_symbol

L'hunk symbol contiene i nomi dei simboli esterni definiti nei file eseguibili (ma anche oggetto). Serve in fase di debugging, anche se non è rarissimo trovare dei file rilasciati, che lo contengano, specie nel PD, quando non ci si è premurati di indicare NODEBUG fra le opzioni del linker. Il formato è il seguente:

```
00033   000003F0   hunk_symbol
00034   00000003   lunghezza del nome in longword
00035   70735F63   nome: "ps_common" con zeri finali
00036   6F6D6D6F
00037   6E000000
00038   00000030   offset nell'hunk in cui compare
                  la definizione
```

Dopo l'identificatore compare la prima "unità dati del simbolo" composta dalla lunghezza del nome in longword e seguita dal nome (in questo caso "ps_common") che, a sua volta, viene seguito da degli 0 fino alla conclusione della longword. Dopodiché, compare l'offset in cui si trova il simbolo.

E' chiaro che un hunk di questo tipo non viene caricato dal loader in memoria e che l'offset si riferisce all'hunk code, data o bss cui è associato. La fine dei simboli è segnalata dalla presenza di una longword a 0. Si noti che la struttura di questo hunk è la stessa degli hunk_ext che compaiono nei file oggetto e di cui parleremo a suo tempo.

hunk_overlay

Se il file ha una struttura ad overlay, al termine del nodo primario comparirà un hunk_overlay ($3F5) che indica al loader di lasciare aperto il file, dopo aver caricato il primo nodo, per il programma detto "overlay manager" o supervisore d'overlay. Il codice corrispondente è contenuto nel file eseguibile e vi è inserito da ALINK (ultimamente, anche da BLINK); è anche possibile scrivere un proprio supervisore d'overlay che compia lo stesso lavoro e inserirlo come prima routine nel primo hunk_code del nodo primario.

Il formato dell'hunk_overlay prevede due longword iniziali: la prima indica la lunghezza della tavola di overlay, la seconda il massimo livello d'overlay maggiorato di 2. Segue la tavola utilizzata dal manager. Dopo questo hunk possono comparire nel file altri hunk_header che indicano l'inizio di un nuovo nodo.

hunk_break

E' costituito da una sola longword ($3F6) che segnala la conclusione di un nodo d'overlay.

Il programma su disco e la prossima puntata

Terminiamo qui l'analisi dei file eseguibili e del mondo del loader. La prossima volta ci addentreremo nell'universo più esoterico dei file oggetto: nei misteri delle librerie linked, dei nomi degli hunk, dei riferimenti e delle definizioni simboliche; nel territorio, insomma, in cui domina il linker e i suoi riti.

Nel frattempo divertitevi a dissezionare i file di Amiga con il programma ARexx presente su dischetto e chiamato hunk.rexx, eccone il formato:

```
rx hunk file [RELOC] [SYMBOL] [DEF] [LATTICE]
```

L'unico parametro obbligatorio è il nome del file da analizzare. Se non indicate nessuna keyword, verranno visualizzati tutti gli hunk del file eseguibile (anche quelli in overlay) o del file oggetto con le informazioni essenziali. Le diverse keyword obbligano il programma a emettere delle informazioni ulteriori, secondo lo schema qui indicato:

(segue a pag. 50)

Dai Lettori: Magic_Square! e...

Tecniche di programmazione

Fabio Oliva

Riceviamo con vivo piacere, dopo l'appello lanciato sul numero 21 di Amiga Magazine-Amiga TransAction, un articolo di un nostro lettore. Ringraziando Fabio Oliva, speriamo che molti altri articoli e programmi giungano in redazione.

MagicSquare è il frutto dello studio e applicazione di particolari tecniche di programmazione in ambiente Workbench standard. Lo scopo di questo articolo è di evidenziare tutte le tecniche e relative problematiche incontrate nello sviluppo di questo programma. In particolare, potrete apprendere come utilizzare la funzione DrawImage di Intuition, che consente di tracciare immagini in una finestra standard associata ad una struttura RastPort.

Apprenderete come effettuare un modesto scroll con l'ausilio della funzione ScrollRaster e dell'evento IDCMP (Intuition Direct Communications Ports) INTUITICKS. E molti altri piccoli accorgimenti utili per la convivenza con Intuition e l'ambiente standard Workbench. Inoltre, potrete passare direttamente dalla teoria alla pratica, analizzando il sorgente di Magic_Square! esaurientemente commentato.

Come creare immagini con la funzione DrawImage

Poter utilizzare immagini create con un normale programma di disegno all'interno dei propri programmi, conferisce sempre un effetto di professionalità non indifferente. Per ottenere tutto ciò, occorre saper approntare una struttura Image e saper usare la funzione DrawImage di Intuition.

Questo è l'unico modo se si vuole evitare di utilizzare metodi non standard e quindi "Illegali" perché difficilmente compatibili con diverse configurazioni o nuove release del sistema operativo.

Il primo passo è quello di creare un disegno con un normale programma tipo DeluxePaint; inscrivetelo in un rettangolo di larghezza uguale a un numero multiplo di otto o sedici. Ciò consentirà di risparmiare qualche manciata di byte e, inoltre, ci faciliterà molto le operazioni seguenti. La lunghezza del disegno è del tutto a vostro piacimento, ma dovrete porre particolare attenzione nell'uso dei colori: vi consiglio di "simulare" la palette dove verrà creata l'immagine, utilizzando gli stessi colori nella stessa identica sequenza. Cioè ogni registro colore dello schermo dove disegnate, dovrebbe avere la stessa tonalità di colore dello schermo in cui creerete l'immagine. Questo vi eviterà di creare un'immagine con un effetto cromatico diverso da quello previsto.

Terminata la fase di disegno, salvate l'immagine come brush di dimensioni esattamente uguali a quelle del vostro rettangolo. Vi ritroverete ora con un file IFF contenente i dati della vostra immagine.

Il secondo passo consiste nel trasformare i dati del chunk BODY del vostro file IFF, in un vettore di word accettabile dal vostro compilatore. Esistono molte utility P.D. per ottenere dati grezzi direttamente dai file IFF, io ne ho utilizzata una creata da un mio amico che accetta un file IFF e crea un file ASCII con tutti i dati dei Bitplane.

Oppure potete usare un metodo laborioso quanto artigianale:

1) Far risalvare il file IFF in formato NON compresso (ad esempio, PixMate lo permette).

2) Eseguire un "type opt h" ridirigendo l'output in un file temporaneo (per esempio, type >ram:Dati Immagine.pic opt h).

3) Con un buon text editor (come Cygnus Ed) caricate il file temporaneo e a forza di taglia e incolla colonnari, isolate tutto ciò che segue il chunk "BODY".

Eliminate ora gli offset sulla sinistra e la rappresentazione ASCII sulla destra. Eliminate i dati relativi alla parola BODY e le 2 Long Word (otto caratteri ASCII) successive. Trasformate questi dati in un vettore di WORD in notazione esadecimale accettabile dal vostro compilatore. Avrete finalmente ottenuto i dati dei Bitplane del vostro disegno.

Purtroppo, questo metodo è valido solo se il vostro disegno è largo un numero multiplo di otto. Se così non fosse, si potrebbe generare un disallineamento che potrebbe risultare fatale per i nostri scopi. Se pensate che tutto ciò sia complesso avrete capito quanto è utile il software PD in

certi momenti. Se avete ancora dubbi su come ottenere i dati grezzi della vostra immagine potete consultare i vari articoli apparsi su Amiga Magazine riguardanti il formato IFF: questo vi permetterà di progettare un piccolo programma adatto a questo scopo.

Il terzo punto consiste nell'approntare una struttura Image adatta alle caratteristiche del vostro disegno e alle caratteristiche dell'ambiente video contenitore. Ecco dunque la struttura in esame che potrete trovare nel file include "Intuiton/Intuition.h" :

```
struct      Image
{
SHORT       LeftEdge;
SHORT       TopEdge;
SHORT       Width;
SHORT       Height,    Depth;
USHORT      *ImageData;
UBYTE       PlanePick,      PlaneOnOff;
struct      Image      *NextImage;
};
```

I primi due elementi indicano la posizione relativa all'angolo superiore sinistro della finestra che dovrà avere l'immagine una volta tracciata, LeftEdge sarà quindi lo scostamento orrizzontale e TopEdge quello verticale.

I tre parametri che seguono indicano rispettivamente la larghezza, altezza e numero di Bitplane necessari alla definizione dell'intera immagine.

Il dato successivo (*ImageData) è un puntatore all'area di memoria in cui sono contenuti dati che servono per la definizione dell'immagine. Con un opportuno Cast assegnategli l'indirizzo base del vettore di WORD creato in precedenza. Se, ad esempio, la vostra immagine è larga 30 pixel, lunga 20 ed è formata da 2 Plane (4 colori) dovrete predisporre due plane IN MEMORIA CHIP formati da 20 righe di 2 WORD (occorre arrotondare la lunghezza della riga alla word successiva ovvero 30 bit arrotondati a 32 e cioè 2 WORD).

I manuali in mio possesso fanno un po' di confusione a proposito dell'organizzazione di questi dati: è bene, quindi, approfondire un po' il concetto.

I Bitplane, nel caso della struttura Image da utilizzare con DrawImage, devono essere posti in sequenza: prima si esaurisce la definizione di tutto il primo, a cui deve seguire quella di tutto il secondo e così via fino all'ultimo Bitplane. La definizione di ogni singolo Bitplane deve essere così intesa: si esaurisce la definizione di tutte le H righe complete formate da N/16 WORD in maniera sequenziale; dove H rappresenta l'altezza dell'immagine e N è la larghezza dell'immagine arrotondata ad un multiplo di 16 bit. I due seguenti parametri, PlanePick e PlaneOnOff, sono molto utili nel caso di immagini che utilizzano un numero di BitPlane inferiore a quello della finestra che conterrà l'immagine. PlanePick, infatti, informa il sistema su dove devono venire inseriti i Bitplane definiti nella struttura Image. Dovremo settare ad uno i bit di questo parametro corrispondenti ai plane che riceveranno i dati della struttura Image. Se, per esempio, PlanePick vale 5 (binario 00000101) i Bitplane zero e due riceveranno i dati della struttura Image.

Il parametro PlaneOnOff decide come devono essere modificati i Bitplane che non ricevono direttamente dati tramite PlanePick. Se, ad esempio, il bit relativo ad un Bitplane che non è interessato tramite PlanePick al ricevimento del bitplane è impostato ad uno, tutti i bit di quel bitplane saranno posti ad uno al momento del tracciamento dell'immagine.

Pensiamo, dunque, ad una Bitmap destinazione con 5 Bitplane e ad un'immagine formata da un solo BitPlane. Impostando il parametro PlanePick ad uno dei cinque valori possibili, si traccerebbe un'immagine che può assumere solo cinque tra i 32 colori consentiti dai 5 BitPlane (ovvero quelli associati alle potenze del due: 2,4,8,16,32). Modificando il parametro PlaneOnOff è possibile usare un colore fra i 32 possibili perché anche gli altri quattro Bitplane riceverebbero una modifica adeguata alle nostre esigenze.

Con il parametro NextImage è possibile definire una lista di immagini da tracciare con una sola chiamata a DrawImage; si deve però tener presente che DrawImage è distruttiva nei confronti dei bitplane non interessati da PlanePick: PlaneOnOff uniforma tutto la Bitmap al valore contenuto in esso. Ciò che vorrei sottolineare è che i dati puntati dal parametro *ImageData DEVONO essere contenuti in CHIP RAM. Questo si può ottenere con specifiche opzioni di compilazione o con le immancabili utility PD. Occorrerà dunque specificare che il segmento DATA del codice eseguibile sia caricato in CHIP RAM.

Il quarto punto si esaurisce nella opportuna chiamata alla funzione DrawImage della libreria Intuition. Esaminiamo i parametri richiesti:

```
void  DrawImage(struct RastPort *RPort,struct
Image *Image,long LeftOffset,long TopOffset);
```

Il parametro *RPort non è altro che un puntatore alla struttura RastPort associata alla finestra ove vorrete creare l'immagine.

Il parametro *Image è un puntatore alla struttura Image creata e inizializzata appositamente.

I parametri LeftOffset e TopOffset sono offset che, aggiunti a quelli nella struttura Image, consentono di usare la stessa struttura Image per tracciare più immagini in locazioni differenti. Ora non vi resta altro da fare che aprire la libreria

Intution e effetture un'opportuna chiamata alla funzione DrawImage.

Intuition darà il compito al Blitter di creare l'immagine nell'esatto punto prefissato. Se per errore le misure dell'immagine eccedono quelle della finestra, l'immagine verrà troncata senza effetti controproducenti. Se per caso i dati puntati da *ImageData non saranno allocati in CHIP RAM, l'immagine verrà creata o completamente trasparente o "sporca".

Scroll e INTUITICKS

Una volta realizzata un'immagine di questo tipo, può essere molto interessante dotarla di una piccola "finestrella" dove viene visualizzato un messaggio attraverso un semplice scroll. Generare un effetto di questo tipo su Amiga, è molto semplice grazie ai particolari chip custom come il Blitter (da BLock TransferER) che è stato progettato per facilitare proprio questo tipo di operazioni. Progettando il nostro scroll occorre tenere presente alcune specifiche, ecco le più importanti:

1) Dimensione della striscia di scroll.

2) Metodo utilizzato per lo shift (scorrimento) dei dati.

3) Temporizzazione.

Per striscia di scroll si intende l'area dove il testo da scrollare verrà mantenuto in movimento. Le dimensioni della striscia di scroll sono, quindi, in stretta relazione alle dimensioni del font che si vuole utilizzare. In particolare l'altezza deve essere uguale o superiore all'altezza del testo.

Occorre considerare anche che alcuni segni come le virgole o come, ad esempio, una "p" minuscola hanno dei segni discendenti e possono invadere una porzione di spazio non occupata dai normali caratteri. Occorrerà anche lasciare qualche pixel di intermezzo per dare "respiro" al testo. Naturalmente più è vasta la dimensione della finestra di scoll e più pesante sarà il lavoro di shift dei dati da parte delle routine incaricate.

Per Metodo di Shift si intende il sistema usato per ottenere il periodico scorrimento del testo all'interno della striscia di scroll. L'efficenza di questo metodo deve essere direttamente proporzionale alle dimensioni della striscia di scroll. In pratica ciò che ci serve è una routine che effettui lo shift in una precisa direzione di una porzione di Bitmap definita da noi. La libreria Graphics ci viene ancora una volta in aiuto con la funzione ScrollRaster.

```
void  ScrollRaster(struct  RastPort
*rastPort,long  dX,long  dY,long minx,long
miny,long maxx,long maxy);
```

Questa funzione è molto versatile ed efficace perché fa uso diretto dell'onnipotente Blitter; vediamo ora come gestire i suoi parametri. Il primo parametro è un puntatore alla struttura RastPort di controllo della nostra finestra checontiene la "striscia di scroll".

I due parametri seguenti indicano lo spostamento rispetto agli assi x e y che dovrà subire la Bitmap indicata dalla struttura RasPort. Per esempio, se si vorrà ottenere uno spostamento della Bitmap di 3 pixel a destra e di 1 pixel verso l'alto, i parametri dX,dY assumeranno i valori -3,+1. Il valore zero indica nessun spostamento in quell'asse. I quattro parametri seguenti minx,miny maxx,maxy non sono altro che le coordinate dei due punti superiore sinistro e inferiore destro che rappresentano il rettangolo che subirà lo spostamento.

L'area liberata dallo spostamento, verrà riempita dalla Graphics library con il colore della Penna di BackGround (BgPen). I pixel che si trovano agli estremi dell'area nel momento dello spostamento verranno "buttati fuori" senza influenzare l'area circostante.

Il metodo di temporizzazione rappresenta il cuore vero e proprio del nostro scroll; la sua efficenza sarà direttamente proporzionale alla fluidità e pastosità dello scroll. Ricordate che non potete fare a meno di un sistema di temporizzazione: in un sistema multitasking come Amiga utilizzare la tecnica di "polling" o ritardo è una grave scorrettezza. Sprecare ingenti quantità di CPU time per un banale ciclo di ritardo è quindi da evitarsi con ogni mezzo. Per venire incontro alle nostre esigenze Intuition mette a disposizione un particolare evento: l'evento IDCMP INTUITICKS.

Settando il parametro IDCMPFlags della struttura NewWindow, al momento dell'apertura della finestra, si potrà ricevere ogni 5 tick (un tick equivale a 1/50 di secondo) un messaggio di Intuition che potremo utilizzare per i nostri scopi di temporizzazione.

E' importante sottolineare che i messaggi ci verranno inviati solo se la finestra e attivata e che Intuition non ci invierà più messaggi finché non avremo risposto a tutti quelli già pervenutici in precedenza: questo per evitare che si possa inutilmente riempire la memoria di messaggi inutilizzati. Naturalmente, questa soluzione è tanto utile quanto improvvisata. Essa si adatta a requisiti di facilità di implementazione gestione penalizzando al contempo la flessibilità del sistema. Inutile dirlo: chi avesse bisogno di temporizzazioni più precise o personalizzate dovrà necessariamente utilizzare il device Timer con tutti i vantaggi e complicazioni che questo comporta.

Ora disponiamo di tutti gli elementi necessari per procedere all'assemblaggio delle nostre routine. Primo passo necessario è quello di aprire le librerie Intuition e GfxBase. Apriamo una finestra (che avrà il parametro IDCMPFlag inpostato a INTUITICKS) che conterrà la nostra "striscia di scroll" e tracciamo una eventuale immagine tramite DrawI-

mage. Ora dovremo predisporre un ciclo di attesa (tramite Wait) di messaggi IDCMP.

Consultando la UserPort, tramite GetMsg, se abbiamo ricevuto un messaggio, potremo stabilire con un semplice switch se si tratta di un evento INTUITICKS. In caso affermativo dovremo provvedere a stampare un singolo carattere nell'angolo destro, effettuare uno scroll di tutta la striscia verso sinistra per un tratto corrispondente alla larghezza del font usato. Naturalmente, vi consiglio di suddividere il movimento in varie fasi, in modo da rendere il tutto più fluido. Se il vostro scroll è molto veloce, potrete incontrare tra gli altri qualche problema di sincronizzazione.

E' un caso a dir poco sporadico, ma è possibile che proprio mentre il sistema effettua lo shift della Bitmap, il raggio catodico stia per passare proprio in quel punto generando un effetto di spezzettamento: ad esempio, può accadere che la parte superiore del disegno si trovi spostata rispetto a quella inferiore. Per evitare tutto ciò, è necessario effettuare i cambiamenti al quadro video in concomitanza con il Vertical Blank, ovvero quando il raggio catodico è giunto nell'angolo inferiore destro e sta risalendo in alto a sinistra. E' possibile ottenere tutto ciò con le funzioni WaiTOF e/o WaiTBOVP della libreria Graphics.

WaitTOF non richiede parametri e attende semplicemente l'interrupt del Vertical Blanking prima di cedere il controllo al task. WaitBOVP richede come parametro un puntatore alla struttura ViewPort contenente la vostra finestra. Essa attende che il quadro della ViewPort sia stato completamente tracciato prima di cedere il controllo al task. Detto questo dovete sapere che trovare la giusta temporizzazione, velocità e fluidità di scroll, richiede un certo numero di tentativi. Dovrete ponderare sapientemente i tempi prima di ottenere risultati apprezzabili. Ma vi garantisco che spesso il risultato vale il tempo perso.

Bug & Tricks

Vorrei ora parlare dei bug che ho dovuto stanare e scacciare dalle righe del mio sorgente. Queste piccole osservazioni sono il frutto del mio lavoro. Spero che così facendo il lavoro da me svolto possa essere utile anche a voi. Con il nuovo S.O. 2.0 lo schermo Workbench può assumere configurazioni molto differenti da quelle a cui ci eravamo abituati. Se avete bisogno di conoscere le dimensioni dello schermo WBench potete utilizzare la funzione GetScreenData.

Questa creerà una copia della struttura Screen specificata che potrete consultare per avere tutte le informazioni necessarie. Se volete sapere se il vostro programma sta girando su una macchina PAL o NTSC vi consiglio di consultare il parametro VBlankFrequency della struttura ExecBase: se la macchina è PAL questo paramentro conterrà 50 (ovvero il numero di volte in un secondo in cui il quadro video viene rigenerato), altrimenti conterra il valore 60 e la macchina sarà NTSC. Non esistono funzioni che generano numeri assolutamente casuali; le funzioni che utilizzano un metodo matematico (ad esempio, RangeRand delle librerie Manx) spesso hanno qualche controindicazione.

Spesso queste funzioni hanno la tendenza ad impoverire il seme in modo tale che dopo diverse migliaia di chiamate generano solo pochi numeri dell'intervallo richiesto. Si può risolvere questo problema espandendo l'intervallo dei valori moltiplicandolo per un opportuno coefficente, per poi dividere il risultato della funzione per lo stesso coefficente.

Può anche capitare che le sequenze siano sempre identiche. Ad esempio, le prime dieci chiamate generino sempre gli stessi 10 numeri indipendentemente dalle condizioni del sistema. Si può risolvere il problema facendo riferimento all'unico elemento sicuramente imprevedibile in una macchina: l'uomo che la controlla. Ad esempio, potete controllare la posizione del puntatore in una situazione particolare e ricorrente, e succesivamente effettuare tante chiamate alla funzione in relazione alla coordinata x del puntatore. Solo la vostra fantasia pone dei limiti in questo campo.

Io odio coloro che modificano le stringhe ASCII dei vari Copyright all'interno di un programma e si attribuiscono la paternità di quel codice. Si può evitare tutto ciò "cripttando" le stringhe più significative tramite delle semplici operazioni a livello di Bit. Anche un semplice Not può fare miracoli in questo senso. Potete anche limitarvi a controllare tramite un checksum l'integrità del messaggio, ma tutto diventa un po' più complicato. Usate manciate di BitShift, Xor, Nand ecc. come rimedio a queste basse forme di pirateria.

Quando utilizzate delle funzioni della libreria Intuition in stretta sequenza, è probabile che qualcosa non vada per il verso giusto. Per risolvere l'inconveniente provate ad interporre un piccolo ritardo tramite Delay tra le successive chiamate. Tutto questo si rende necessario per far sì che Intuition possa aggiornare le sue liste di sistema prima che gli venga chiesto di eseguire un altro comando. Se, per esempio, noi chiediamo di aprire una finestra e immediatamente dopo di tracciare qualcosa in essa può accadere che il secondo compito non venga eseguito perché il sistema ancora non possiede i dati correttamente aggiornati del primo evento. Chi utilizza i codici oggetto forniti con alcuni dei più noti compilatori per generare codici Stay & Run Resident (detach.oXX -> Manx , CBack.o -> Lattice) ponga particolare attenzione al parametro *_procname. Questo parametro deve puntare alla stringa che identificherà il processo residente da creare. La stringa di identificazione del processo DEVE essere diversa dal nome che utilizzerete per salvare l'eseguibile. Se così non fosse vedreste apparire un bel Guru che indica "Lista di memoria danneggiata". Tutto questo perché i processi vengono riconosciuti tramite questa stringa e un omonimia creerebbe un po' di confusione al gestore dei task (Exec).

Intelligenza artificiale e risoluzione automatica dei problemi

Una breve introduzione (parte I)

Alberto Geneletti

L'intelligenza artificiale è una disciplina che si occupa del progetto e della realizzazione di programmi intelligenti sui calcolatori elettronici. In realtà il calcolatore rappresenta soltanto una necessità pratica, lo strumento più appropriato ed efficiente che la tecnologia attuale mette a disposizione per la costruzione di intelligenze che funzionino in modo automatico, senza l'intervento dell'uomo.

E' prevedibile che in futuro esso venga rimpiazzato da qualche altra macchina più adeguata, che permetta di superare gli ostacoli insormontabili nei quali questa disciplina sembra essersi arenata dopo i primi entusiasmanti successi, che si sono però limitati solo a qualche applicazione particolare, come la risoluzione automatica dei problemi.

In questo articolo prenderemo proprio in considerazione alcune delle tecniche sviluppate per questa applicazione specifica, e vedremo come sia possibile implementarle su Amiga, un computer che in questo campo non ha ancora avuto modo di segnalare la propria presenza.

Programmi intelligenti

Prima di tutto, occorre precisare che cosa si intende per programma intelligente. Assumeremo, quindi, come primitivo il concetto di intelligenza, che risulterebbe difficile definire senza rimanere nel vago, e diremo che un programma è intelligente quando è in grado di fornire prestazioni che riterremmo frutto di intelligenza qualora fossero esibite dall'uomo.

In questo senso, risulta irrilevante simulare il modo di pensare dell'uomo, perché il giudizio è limitato ai risultati conseguiti; il principio di funzionamento del programma può quindi essere ben diverso da quello operante a livello cerebrale, e del resto non può che essere così, dal momento che non abbiamo ancora risolto la maggior parte dei misteri nei quali sono avvolti i nostri processi mentali.

Questa assunzione ci permetterà di non perderci in divagazioni filosofiche, che pur essendo di competenza di una disciplina conoscitiva come l'intelligenza artificiale, porterebbero il nostro discorso troppo lontano dalle nostre intenzioni.

Risoluzione automatica dei problemi

La risoluzione di problemi è un tema dominante nel campo dell'intelligenza artificiale. Nel significato più ampio, questa applicazione potrebbe riguardare tutta l'informatica, poiché anche l'esecuzione meccanica di calcoli può essere vista come risoluzione di un problema.

E' invece di competenza dell'intelligenza artificiale quella classe di problemi ai quali non sono applicabili gli abituali metodi di calcolo, per i quali, cioè, non è definita a priori una formula risolutiva, come accade, invece, nel caso di un sistema di cinquanta equazioni o della derivazione di funzioni; per tali problemi, infatti, non occorre intelligenza, ma solo una buona dose di pratica e di attenzione.

Dei problemi di cui ci occuperemo invece, anche se è possibile avere un'idea generale di come sarà la soluzione, non è noto il processo risolutivo, che dovrà essere ricavato per tentativi.

In questo caso risulta di primaria importanza la capacità del risolutore di apprendere dalla propria esperienza e dai propri errori, riducendo ad astrazioni generali tanto i tentativi che hanno portato alla soluzione, quanto gli immancabili insuccessi.

In realtà, non ci sono per il momento elementi sufficienti per affermare che un sistema dotato di un tal livello di autocoscienza, potrà essere realizzato. La risoluzione automatica dei problemi, è molto più frequentemente limitata alla ricerca per tentativi nello spazio di tutte le possibili soluzioni.

Questi tentativi possono essere effettuati alla cieca, seguendo soltanto un criterio che regola l'ordine nel quale verranno generati, oppure potranno essere condizionati da una funzione di valutazione, che sia almeno in grado di farci sapere se ci stiamo avvicinando o meno alla soluzione.

Quest'ultimo metodo di ricerca è noto con il nome di RICERCA EURISTICA. Il termine "euristico" significa infatti "utile alla ricerca", ed è attribuibile ai criteri utilizzati dalla funzione di valutazione, che non è in grado di fornire stime

di carattere assoluto, ma può soltanto aiutare a trovare la soluzione in modo più rapido, evitando, ad esempio, di effetture tentativi poco promettenti e di concentrarsi su quelli che lo sono maggiormente.

Esempi di problemi

L'oggetto di studio preferito dall'intelligenza artificiale, è stato fin dall'inizio quello della risoluzione di giochi intelligenti.

Fanno parte di questa numerosa famiglia di giochi alcune celebrità, come la dama e gli scacchi, e un gran numero di altri passatempo importati in occidente dalla cultura e dalla tradizione cinese, come il Kalah e il Go; ricordiamo, inoltre, il cubo di Rubik, la torre di Hanoi e il gioco del quindici, il rompicapo per il quale abbiamo sviluppato un'implementazione in Modula-2 dei tre metodi di ricerca di cui ci occuperemo.

Per chi non lo sapesse, il gioco del quindici consiste in un telaio di quindici tessere numerate, disposte in una cornice quadrata 4 x 4; rimane così libera una casella, che permette alle altre possibilità di movimento.

Le tessere possono così essere mischiate, assumendo una configurazione disordinata: scopo del gioco, è ricostruire la sequenza di partenza, nella quale le caselle si trovano ordinate in ordine crescente, con la cella vuota in basso a destra.

Questo gioco, come del resto tutti gli altri sopracitati, si presta particolarmente bene ad essere implementato su un calcolatore, perché è un problema che può essere definito in termini precisi e rigorosi.

Esiste, infatti, una configurazione iniziale e una finale, alla quale bisogna pervenire applicando un certo numero di regole, che chiameremo OPERATORI.

Purtroppo, non tutti i problemi reali possono essere definiti con altrettanta precisione (basta pensare al problema di aver successo con le donne!); esiste cioè un gran numero di problemi che non sono suscettibili di una facile rappresentazione formale.

Il primo approccio resta, comunque, quello di cercare di rappresentare il problema per mezzo di un numero finito di simboli, che devono poter essere manipolati dagli operatori, fino ad assumere una configurazione particolare che stabiliamo costituire la soluzione del problema.

Nel caso del gioco del quindici, tale configurazione corrisponde ad una disposizione delle tessere ordinata in modo particolare, mentre gli operatori non sono altro che le quattro possibili mosse, nelle quattro direzioni principali, della casella libera di muoversi.
Naturalmente, per raggiungere la disposizione finale, sarà necessario passare attraverso un certo numero di altre configuarazioni, che chiameremo STATI del problema, seguendo la terminologia tradizionale.

La totalità degli stati costituisce lo SPAZIO DEGLI STATI, nel quale le varie configurazioni possono essere legate una all'altra per mezzo di un operatore.

Questo significa che è possibile passare da uno stato a quello adiacente applicando ad esso un operatore, e che nello spazio degli stati esistono infiniti percorsi che legano tra loro configurazioni anche molto diverse e lontane.

La ricerca della soluzione consiste proprio nel determinare il tragitto che collega la disposizione finale a quella iniziale.

Non è detto poi che questo sia sempre possibile: nel gioco del quindici, ad esempio, si può dimostrare che lo spazio degli stati è costituito da due insiemi disgiunti. Gli elementi di ciascun insieme sono tutti collegati tra loro, ma non esiste nessun ponte di trasformazione di uno stato appartenente al primo in uno del secondo.

Chi non ci credesse provi a verificare che non è possibile raggiungere la disposizione finale:

```
15 14 13 12

11 10  9  8

 7  6  5  4

 3  2  1  .
```

Implementiamo stati e operatori

La scelta del Modula-2 come linguaggio di programmazione si è come al solito rivelata particolarmente adeguata.

Non solo per l'alto grado di leggibilità, caratteristica che ha reso il Modula-2 il linguaggio didattico per eccellenza, ma nel nostro caso, soprattutto, perché ci ha offerto la possibilità di ridefinire tipi di dati, per mezzo del costruttore TYPE.

E' stato così possibile dare una rappresentazione semplice ed efficace agli stati del nostro rompicapo, definendo il tipo Stato come ARRAY bidimensionale 4 x 4 di NumeroTessera, un sottoinsieme dei CARDINAL limitato da 0 a 15.

Ai quattro operatori:

InAlto, InBasso, VersoDestra, VersoSinistra

è stata, invece, fornita come implementazione una procedura booleana, che restituisce FALSE in caso di mancata applicabilità dell'operatore. Va infatti osservato che non sempre è possibile applicare tutti quattro gli operatori ad un determinato stato; quando la casella vuota si trova in pros-

simità del bordo del telaio potranno essere applicati solo tre operatori diversi, quando si trova in un angolo solo due.

Vediamo in dettaglio come funzionano queste procedure. Innanzitutto gli identificatori si riferiscono alla direzione nella quale verrà spostata la casella vuota all'interno dello stato s, che viene passato come parametro per indirizzo.

Questo significa che la procedura trasformerà direttamente il parametro effettivo, e non una sua copia, e che quindi otterremo come risultato dell'esecuzione, oltre al successo o all'insuccesso di cui parlavamo, anche lo stesso stato di partenza trasformato in modo opportuno.

Occorre, quindi, determinare la posizione della casella vuota all'interno dello stato s. Questa operazione, comune ai quattro operatori, viene effettuata da un'altra procedura, la CercaCasellaVuota, che restituisce in X e in Y le due coordinate richieste.

Dopo aver controllato l'applicabilità dell'operatore (nel caso di InAlto si verifica, ad esempio, che la casella libera non si trovi nella prima riga) vengono scambiate tra di loro le posizioni della casella libera e di quella al essa adiacente nella direzione specificata.

Sono poi stati definiti alri due operatori: Mischia, che si occupa di far assumere allo stato parametro una disposizione disordinata casualmente, e Ordinato, procedura booleana che restituisce TRUE quando lo stato parametro si trova nella configurazione ordinata (successione crescente dall'1 al 15 e casella libera in basso a destra).

In un'implementazione più generale degli operatori applicabili ad un problema, sarebbe poi necessario associare a ciascuno di essi una funzione di costo, dal momento che all'applicazione di strategie differenti possono corrispondere costi diversi nel mondo reale.

Nel gioco del quindici questo non è necessario: a tutti gli operatori può essere attribuito costo unitario, e in questo modo il costo totale della soluzione è rappresentato dal numero di operatori che è necessario applicare per raggiungere la disposizione finale.

Alcuni degli algoritmi di ricerca di cui ci occuperemo, permettono di ricavare proprio la soluzione a costo minimo, fra le tante possibili alla risoluzione di un problema nello spazio degli stati, quella che comporta cioè il minor numero di mosse.

In questo modo, risulterà interessante confrontare le prestazioni umane con quelle dell'intelligenza artificiale, che pur richiedendo tempi di calcolo più lunghi di quelli di una risoluzione umana, alla fine della ricerca ostenteranno nella maggioranza dei casi una soluzione migliore della nostra.

La sfida è quindi aperta!

Organizzazione degli stati in memoria

Da quanto è stato detto, può sembrare che condizione necessaria alla ricerca della soluzione sia la disponibilità in memoria dell'intero spazio degli stati. Questo è tuttavia assurdo, perché le sedici tessere possono essere disposte in ben 200.000 miliardi di configurazioni diverse.

Nella prima casella possiamo infatti portare una qualsiasi delle 16 tessere, nella seconda tutte le altre tranne quella, nella terza una qualsiasi delle 14 rimanenti.

Con solo tre caselle otterremmo così 16x15x14 configurazioni diverse.
Ripetendo poi lo stesso ragionamento per tutte le tessere si ottiene il valore:
16x15x14x13x12x11x10x9x8x7x6x5x4x3x2x1,
detto anche 16 fattoriale, che corrisponde appunto al valore menzionato poc'anzi.

Per ovviare all'inconveniente della disponibilità di memoria, gli algoritmi di risoluzione automatica costruiscono, poco per volta, una porzione sempre più estesa di tutto lo spazio degli stati, in certi casi solo quella porzione che sembra più promettente nei confronti della presenza della soluzione.

Il processo termina quando viene generato lo stato finale, dopo aver richiesto una quantità di memoria non determinabile a priori.

Non è quindi possibile allocare tutta la memoria prima di incominciare la ricerca, e neanche in base ad una stima approssimativa, poiché questo significherebbe scarsa efficenza in un sistema multitasking come Amiga, nel quale più programmi possono aver accesso contemporaneamente alle risorse hardware. Una soluzione plausibile, risulta allora quella di allocare poco per volta solo la memoria necessaria a memorizzare lo stato appena generato. E' necessario collegare poi tutti questi frammenti per mezzo di puntatori, costruendo in questo modo una lista.

La soluzione delle liste è dettata anche da altre ragioni pratiche. Avremo, infatti, bisogno di mantenere separati gli stati ancora inespansi da quelli già generati, di gestire così due liste, che chiameremo rispettivamente OPEN e CLOSED, e di procedure che ci permettano di trasferire uno stato da una lista all'altra.

Oltre alla disposizione delle tessere e al puntatore della lista, avremo bisogno di memorizzare, per ogni stato, anche il puntatore a quello che l'ha generato, la profondità, cioè il numero di stati che lo collegano alla configurazione iniziale o radice, e il valore assunto dalla funzione di valutazione della ricerca euristica.

Per questo abbiamo definito la struttura Nodo della lista nel modo seguente:

```
Nodo = RECORD
              NodoSuccessivo : Link;
                     Matrice : StatoCompresso;
              NodoGenitore : Link;
                Profondita : CARDINAL;
      ValutazioneEuristica : CARDINAL

      END;
```

dove Link è un puntatore a Nodo.
E' stato poi necessario definere anche la struttura:

```
Lista = RECORD
                PrimoNodo : Link;
               UltimoNodo : Link
       END;
```

che mantiene le informazioni necessarie alle seguenti procedure di gestione delle liste.

Gestione delle liste

Come avrete notato, il campo Matrice non corrisponde ad uno stato, come invece avreste potuto aspettarvi; all'interno dei nodi, infatti, la disposizione delle tessere viene compattata, approfittando del fatto che i numeri dallo 0 al 15 possono essere rappresentati con soli quattro bit, cioè mezzo byte.

Uno StatoCompresso è definito allora come un ARRAY [0..3] OF CARDINAL, dove ogni elemento riesce a contenere un'intera riga della matrice delle tessere.

La compattazione e la scompattazione vengono eseguite dalle due procedure Compatta e Scompatta, che accettano entrambe, come primo parametro, lo stato espanso, come secondo quello compresso.

Queste routine, oltre a permettere un risparmio di memoria del 50% per quanto riguarda la disposizione dell tessere e del 35% sulla struttura Nodo, riducono notevolmente i tempi di esecuzione, permettendo di effettuare solo quattro controlli (uno per riga), contro i sedici altrimenti necessari alla procedura NonEsiste, che verifica l'esistenza o meno di uno stato appena generato in una lista, e che è senza dubbio la routine chiamata più frequentemente dagli algoritmi di ricerca.

Ad entrambi i puntatori della struttura lista, viene attribuito al momento dell'inizializzazione il valore NIL, operazione svolta anche dalla procedura AnnullaLista, che necessita di una struttura Lista come unico parametro.

Nel caso la lista in questione sia già stata costruita da una precedente esecuzione di qualche algoritmo di ricerca, e sia così costituita da un numero anche molto elevato di nodi, tale procedura li scarica ad uno ad uno, restituendo al sistema la memoria ad essi dedicata.

In pratica non fa altro che chiamare un'altra procedura, la booleana RimuoviPrimoNodo che permette di estrarre il primo nodo della lista parametro e che restituisce FALSE quando la lista è vuota; proprio questa condizione permette alla procedura AnnullaLista di uscire dal ciclo REPEAT...UNTIL che ne costituisce il corpo principale.

La procedura RimuoviPrimoNodo restituisce oltre a questa informazione tutti i campi significativi della struttura Nodo, e cioè uno stato già scompattato corrispondente alla Matrice, il puntatore al nodo genitore, la profondità e il valore della funzione euristica.

Tutti questi valori vengono memorizzati direttamente nelle variabili omologhe passate per indirizzo alla chiamata, insieme alla struttura Lista dalla quale dovrà essere effettuato il prelievo.

In pratica, viene letto il campo PrimoNodo della struttura Lista, vengono lette le informazioni contenute in questo nodo, si restituisce al sistema la memoria da esso occupata e viene posto nel campo PrimoNodo l'indirizzo del secondo nodo.

Questo non è tuttavia l'unico modo in cui avremo bisogno di estrarre un nodo da una lista. Diversamente dagli algoritmi di ricerca in ampiezza e in profondità, che prelevano per l'elaborazione lo stato in cima alla lista, l'algoritmo di ricerca euristica non si accontenta di un nodo qualunque, ma dà la precedenza a quello euristicamente più promettente; occorre quindi anche una procedura RimuoviNodoMigliore che si informi del valore assunto dal campo ValutazioneEuristica di ogni nodo, e rimuova dalla lista quello per cui tale valore è minimo (il più promettente, come vedremo).

Per far questo, vengono lette, una per una, tutte le valutazioni euristiche, viene ricavato il valore minimo, e ci si muove all'interno della lista grazie ai puntatori NodoSuccessivo fino a ritrovare il nodo la cui valutazione corrisponde al valore minimo.

Viene quindi tagliata la lista, ponendo nel campo NodoSuccessore del nodo che precede quello selezionato un puntatore a quello che lo segue. La procedura prevede anche i casi particolari, come quando vi è un solo nodo nella lista o quando il nodo migliore è anche il primo, casi per i quali viene richiamata la procedura RimuoviPrimoNodo, oppure quando il nodo migliore si trova in fondo alla lista, nel qual caso viene invece modificato opportunamente anche il puntatore UltimoNodo della struttura lista.

Come l'altra procedura di estrazione anche questa restituisce il valore booleano FALSE quando la lista è vuota. Anche per aggiungere nodi alla lista abbiamo bisogno di due procedure di tipo diverso: la prima AggiungiPrimoNodo permette di inserire il nuovo nodo in cima alla lista, la seconda AggiungiUltimoNodo lo pone invece in coda, e

risulterà quindi l'ultimo nodo ad essere rimosso dalla normale procedura di estrazione.

Entrambe richiedono in ingresso la struttura Lista, lo stato da includere nel nuovo nodo e il puntatore da memorizzare nel campo NodoGenitore. AggiungiPrimoNodo si occupa, inoltre, di calcolare il valore del campo profondità e permette di assegnare un valore, che deve essere passato come parametro, anche al campo ValutazioneEuristica.

Alla profondità viene attribuito il valore 0 per il nodo iniziale, e quello della profondità del nodo genitore incrementato di uno a tutti quelli successivi. AggiungiPrimoNodo modifica, inoltre, il campo PrimoNodo della struttura Lista, memorizzando in esso l'indirizzo del nuovo nodo, nel cui campo NodoSuccessivo viene, invece, registrato l'indirizzo del precedente primo nodo, mantenendo così integra la lista.

AggiungiUltimoNodo modifica invece semplicemente il campo UltimoNodo, mentre al campo NodoSuccessivo viene lasciato il valore NIL, indicante proprio che si tratta dell'ultimo nodo della lista.

Per quanto riguarda la gestione delle liste troverete ancora la già citata procedura booleana NonEsiste, che verifica l'esistenza di un determinato stato nella lista, e la procedura ControllaLista, che è stata utilizzata in fase di debugging, e che riteniamo poter essere utile a tutti coloro che intendano cimentarsi in qualche modifica personale del sorgente.

Questa procedura controlla infatti l'integrità di una lista e la presenza in ogni nodo di valori possibili della matrice delle tessere (la somma dei sedici NumeroTessera deve risultare 120), restituendo poi il risultato dell'indagine per mezzo di un messaggio di conferma o di errore.

Nel prossimo numero

Abbiamo ora a disposizione tutto quello che occorre a dare inizio alle nostre ricerche.

Nel prossimo numero vedremo in dettaglio come avviene il processo di ricerca, e prenderemo in considerazione ben tre algoritmi diversi di risoluzione automatica.

L'implementazione in Modula-2 che verrà loro fornita richiederà naturalmente la disponibilità di tutte le procedure definite in questo numero, delle quali troverete il sorgente nella directory TRANSACTION del disco allegato. Daremo, quindi, una valutazione alle prestazioni dei tre metodi di ricerca, e scopriremo come soltanto l'approccio euristico permetta di ottenere risultati in grado di competere con le prestazioni umane.

Infine, potrete verificare voi stessi l'efficacia dei metodi di risoluzione automatica, grazie ad un programma eseguibile che verrà incluso sul disco insieme ai sorgenti, che vi permetterà di confrontarvi con il computer nella risoluzione del nostro rompicapo.

(segue da pag. 36)

I Dispositivi di Input

Sinistra	Y1
Destra	X1

Lo stato del pulsante sinistro del mouse e il pulsante di sparo del joystick possono essere determinati controllando il bit 7 del registro PRA in CIAA. Il registro, lungo un byte, è sito all'indirizzo BFE001.

L'accesso al pulsante destro del mouse è leggermente più complesso, in quanto coinvolge l'uso dei registri proporzionali. Siccome non ho mai avuto bisogno di usarlo, non ho ancora scritto il codice necessario, così dovrete aspettare un articolo futuro per questo tipo di informazioni!

E' tutto per ora, gente. Di più, la prossima volta.

(segue da pag. 41)

Un primo approccio al misterioso mondo degli hunk

RELOC	stampa gli offset delle rilocazioni
DEF	stampa i nomi delle definizioni esterne
SYMBOL	stampa tutti i simboli
LATTICE	stampa i simboli delle librerie Lattice

La keyword RELOC stampa tutti gli offset degli hunk reloc: armatevi di pazienza, proprio oggi ho visto un programma che aveva 974 rilocazioni di un solo hunk rispetto a un altro hunk (ed era composto da 43 hunk complessivi) e l'ARexx non è quel che si dice veloce! La keyword SYMBOL stampa tutti i simboli che trova, compresi quelli degli hunk_symbol. DEF permette di visualizzare i simboli definiti esternamente nei file oggetto. LATTICE serve ad analizzare le librerie indicizzate della Lattice. Alcuni di queste keyword risulteranno più chiare quando parleremo dei file oggetto. Per analizzare con comodità un file di una certa lunghezza conviene ridirigere l'output verso un file ed esaminarne poi il contenuto con MORE o un editor, perché le informazioni stampate sono piuttosto elevate, specie se si specifica qualche keyword opzionale.

Buona esplorazione. E, se scoprite qualcosa di interessante sugli hunk, non esitate a scriverci.

SPECIFICHE DEL PROGRAMMA

CONFIGURAZIONE MINIMA
512K RAM
Kickstart 1.2/1.3

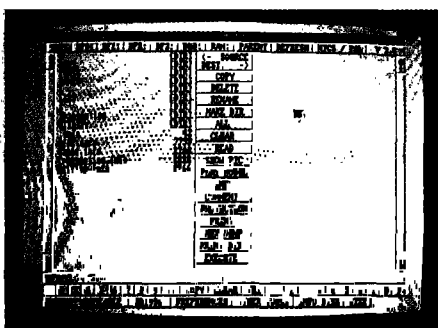
UTILIZZO
Workbench: Doppio click sull'icona
CLI/SHELL:CD Amigamagazine:
utility/AIBB2
Stack 12000
AIBB2

FILE DI SUPPORTO
Arp.library
Req.library

Clockdj

Questo programma combina fra loro varie utility: un orologio, un acceleratore del mouse, un indicatore di memoria libera, inoltre, permette di ridefinire i tasti funzione e di assegnargli delle macro, migliora la gestione delle finestre e degli schermi e molto altro. Per caricare il programma dovete fare un doppio click sull'icona. Subito dopo apparirà una finestra CLI in cui dovrete digitare:

RUN Clockdj

Il programma girerà in multitasking senza disturbare le altre operazioni eseguite dal computer. Per accedere al menu con le funzioni dovete cliccare sulla piccola finestra con l'orologio e poi premere il tasto destro del mouse: al centro dello schermo apparirà il ricco menu del programma. Tutte le funzioni sono molto intuitive e dopo alcune prove padroneggerete il programma alla perfezione. Per i più esperti segnaliamo che il programma è dotato di una porta ARexx accuratamente documentata nel file con le istruzioni che accompagna il programma.

SPECIFICHE DEL PROGRAMMA

CONFIGURAZIONE MINIMA
512K RAM
Kickstart 1.2/1.3

UTILIZZO
Workbench: Utilizzabile solo da CLI/SHELL
CLI/SHELL: CD Amigamagazine:
utility/clockdj
RUN Clockdj

FILE DI SUPPORTO
Arp.library

Diskmaster

Diskmaster è una stupenda utility che permette di avere a portata di mouse la maggior parte dei comandi dell'AmigaDOS: Copy, Delete, Rename, Format, Install e molti altri. Per caricarla fate un doppio click sulla sua icona. Apparirà una finestra CLI in cui dovrete scrivere:

RUN diskmaster

A questo punto apparirà la schermata principale del programma divisa in 5 sezioni: le due aree rettangolari vuote servono a contenere l'elenco dei file delle directory selezionate, la parte centrale del video contiene i comandi principali, la parte superiore contiene l'elenco dei dispositivi a cui si può avere accesso (df0:, df1:, Ram Disk, ecc.), infine la parte inferiore contiene i gadget che permettono di accedere alle preferences e alle utility per la copia e la formattazione dei dischi. Il programma può memorizzare fino a 10 directory diverse, in modo da non dover perdere tempo a rileggere ogni volta il disco. Per selezionare queste directory è sufficiente cliccare col mouse sui gadget numerati da 1 a 10 nella parte bassa dello schermo. Per visualizzare il contenuto di un disco è sufficiente cliccare sul gadget che lo contraddistingue nella parte alta del video.
Per selezionare un file o una directory, basta cliccare su di essi col tasto sinistro del mouse. Cliccando col tasto destro si ha accesso alla directory stessa.
Questo programma è molto utile se si devono copiare molti file o spostare intere directory. Non usando i comandi dell'AmigaDOS il risparmio di tempo è veramente notevole.

SPECIFICHE DEL PROGRAMMA

CONFIGURAZIONE MINIMA
512K RAM
Kickstart 1.2/1.3

UTILIZZO
Workbench: Utilizzabile solo da CLI/SHELL
CLI/SHELL:CD Amigamagazine:
utility/diskmaster
RUN diskmaster

FILE DI SUPPORTO
Nessuno

Sysinfo

SysInfo è una utility che visualizza sullo schermo una enorme quantità di informazioni riguardo al nostro Amiga. Il programma viene fatto partire come al solito con il doppio click sull'icona. Subito dopo il caricamento, in alto a destra, verrà visualizzata la versione di Kickstart, di Workbench e delle librerie principali. In alto a sinistra, invece, viene visualizzata la memoria totale del sistema e la quantità di memoria libera. Le altre indicazioni sullo schermo riguardano la versione dei chip installati e le eventuali schede di espansione. Per quanto riguarda i primi viene visualizzata la versione dei chip Agnus e Denise. In questo caso l'8371 è il vecchio chip Agnus, mentre l'8372 è il nuovo Agnus che può indirizzare fino ad 1 Mb di CHIP RAM. L'8361 è la versione corrente del Denise dato che non è ancora disponibile la nuova versione (8362). Viene anche indicato il tipo di CPU fra 68000, 68010, 68020, 68030 e 68040 e l'eventuale presenza del coprocessore matematico

(68881 e 68882). L'MMU è il chip 68851 presente su alcune schede. Speed in MHz indica la velocità del clock (su un Amiga normale si aggira intorno ai 7.15 MHz). Per quanto riguarda le schede di espansione viene controllata la presenza di Hard Disk o della Bridgeboard (AT o XT). Insieme viene indicato il numero di disk drive collegati. L'ultima operazione che effettua il programma, è il test della velocità. In pratica confronta la velocità di Amiga su cui si sta eseguendo il test con quella misurata su altre configurazioni. Quelle usate per il confronto sono un Amiga 500 con 512 K, un Amiga 2000, un Amiga 2000 con scheda acceleratrice GVP 3001, un Amiga 2500 (in pratica un Amiga 2000 con scheda 68020 a 14 MHz), un Amiga 3000 a 25 Mhz e un PC XT. Questo programma può essere utilizzato per testare il proprio Amiga. Se i dati visualizzati non corrispondono esattamente alla configurazione del computer, possono esserci dei problemi ai chip, alle espansioni di memoria o alle schede montate. Su un Amiga 3000 il programma funziona correttamente solo usando il Kickstart 2.0.

SPECIFICHE DEL PROGRAMMA

CONFIGURAZIONE MINIMA
512K RAM
Kickstart 1.2/1.3

UTILIZZO
Workbench: Doppio click sull'icona
CLI/SHELL:CD Amigamagazine:
utility/sysinfo
Sysinfo

FILE DI SUPPORTO
Nessuno

Mountains

Mountains è un programma per disegnare paesaggi frattali. Anche se è il programma più completo di questo genere e dispone di molte opzioni, è molto facile iniziare a cotruire dei bellissimi paesaggi. Fate partire il programma con il doppio click sull'icona. Appena il programma è caricato selezionate nel primo menu l'opzione Demo. Verrà generato un paesaggio casuale che in seguito verrà riportato in tre dimensioni. Se siete rimasti impressionati dalla demo pensate che con Mountains potete disegnare qualsiasi tipo di paesaggio e osservarlo da una posizione qualsiasi. Inoltre, potete salvare l'immagine in formato IFF in modo da poterla modificare con un programma di disegno qualsiasi o inserirla in uno slideshow o in una animazione.

Le altre opzioni del primo menu sono Interrupt, Save Image, Map Data, Clear All, Undo e Quit. Rispettivamente servono a interrompere il rendering di un immagine, salvare l'immagine, salvare o caricare l'immagine in un formato particolare del programma per successive rielaborazioni, cancellare lo schermo e i dati dell'immagine, annullare l'ultima operazione e uscire dal programma. Il secondo menu, Options, ha sei opzioni: Create, Magnify, Lake, Base, Top e ReScale. Permettono di creare i propri paesaggi, ingrandire porzioni di schermo, stabilire il livello del mare nei paesaggi, stabilire l'altezza minima e massima delle colline, ridimensionare i dati che compongono il disegno. Il terzo menu, Display, ha otto opzioni: Clear, Color, Scale, Frame, Mesh, Sky, Map e 3D Plot. Servono per cancellare il disegno, cambiare la palette dei colori, modificare i parametri delle componenti dell'immagine (terra, acqua e cielo) e tracciare l'immagine in tre dimensioni. L'ultimo menu, Parameters, ha quattro opzioni: Rotation, Magnification, Scales e Size-Light. Con Rotation potete controllare l'angolo e la posizione da cui vedere la scena. Avete a disposizione quattro gadget numerici: X-Angle, Z-Angle, X-Start e Y-Start. I primi due servono a determinare l'inclinazione sugli assi X e Z, mentre gli altri due servono per settare il punto di osservazione. Magnification controlla le dimensioni dell'immagine tramite quattro gadget numerici. Scales controlla l'aspetto dell'immagine tramite il livello di precisione dei frattali. L'ultima opzione, Size-Light, controlla la risoluzione dell'immagine e l'angolo da cui viene illuminata la scena. Il valore di default è 45 gradi che corrisponde circa alle 9:30 del mattino.

SPECIFICHE DEL PROGRAMMA

CONFIGURAZIONE MINIMA
512K RAM
Kickstart 1.2/1.3

UTILIZZO
Workbench: Doppio click sull'icona
CLI/SHELL: CD
Amigamagazine:utility/mountains
RUN Mountais

FILE DI SUPPORTO
Tutti nella directory Mountais

NOTE

- Per decompattare i file con i sorgenti C e Assembler nella directory TransAction utilizzate il programma PowerPacker 2.3b presente sul dischetto allegato al numero 19/20 di Amiga Magazine.
- Nei casi in cui nella tabella con le specifiche del programma alla voce 'file di supporto' è indicato 'tutti nella directory' significa che per copiare il programma su un altro disco o su hard disk è sufficiente, da workbench, trasportare l'icona della directory sul disco desiderato. Se invece è indicato 'nessuno' il programma per funzionare non necessita di altri file. In tutti gli altri casi è necessario copiare oltre alla directory del programma anche i file indicati (ad esempio, librerie, font, device, ecc.).
- Ricordiamo che le librerie arp. library e req.library nella directory LIBS di Amiga Magazine sono di pubblico dominio. Vi consigliamo di copiarle nella directory LIBS del vostro hard disk o del vostro Workbench dato che sono necessarie per il funzionamento della maggior parte del software di pubblico dominio e, quindi, anche di parte del software presente sul disco allegato ad Amiga Magazine.

KCS 3.0: il professionista del MIDI

Questa volta abbiamo provato per voi la punta di diamante della linea Dr.T's: il KCS 3.0, uno fra i più potenti programmi musicali MIDI, integrati, disponibili sul mercato per Amiga.

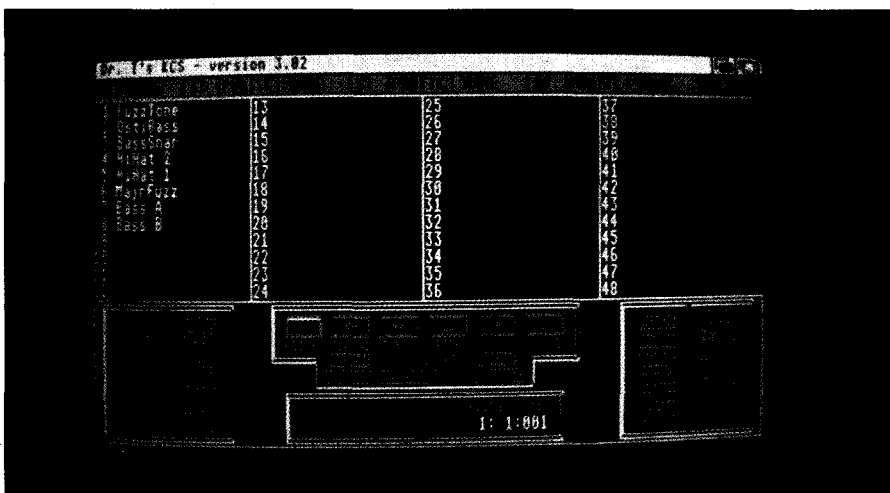
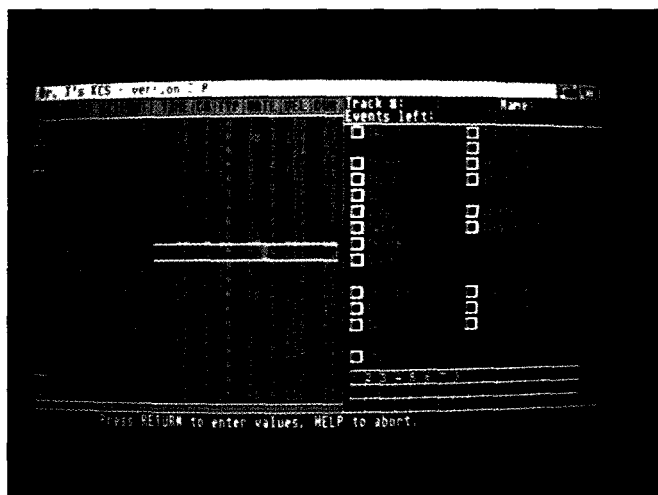
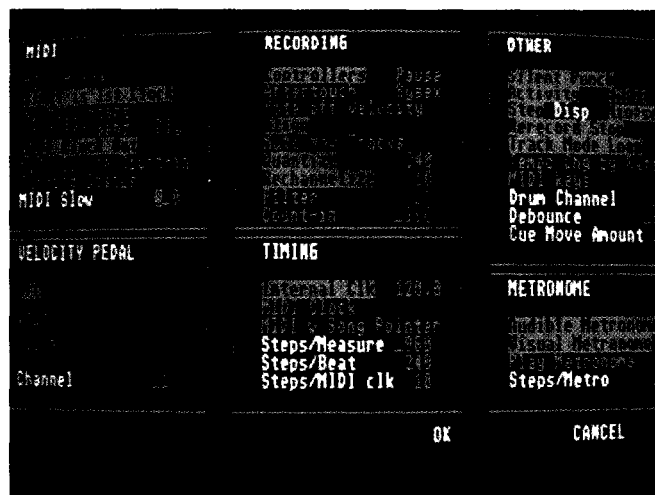
Aldo & Andrea Laus

La precedente versione del KCS, la 1.6, molto conosciuta e abbastanza diffusa tra i pionieri del MIDI su Amiga, ha in comune con la nuova release solo una cosa: il nome.
In effetti il KCS 1.6 si presentava in modo molto altisonante con le sue 48 tracce, delle quali solo 36 visualizzabili contemporaneamente, e con l'editing degli eventi MIDI. Quando si provava ad utilizzarlo, ci si rendeva conto che non era, in fondo, molto potente: difettava dei menu a finestra, non era possibile concatenare una song e i comandi non erano per niente intuitivi.
Questi i motivi per cui non ebbe un successo travolgente, non quello, per lo meno, che ci si aspettava. A qualche anno di distanza, dopo l'uscita di giganti quale, ad esempio, Music-X (vedere articolo in merito sui numeri 11 e 12 '90 di Amiga Magazine), la Dr.T's ha pensato che fosse giunto il momento di dare una veste davvero professionale al suo sequencer; è nato così il KCS 3.0.

Il programma

Il Keyboard Controlled Sequencer V.3.0, è un programma MIDI integrato che si presenta come una compatta workstation di lavoro, in cui è possibile creare brani musicali, curandone ogni particolare, dalla registrazione delle varie tracce, alla miscelazione dei volumi in uscita. Il programma principale si articola attraverso una ampia serie di videate e si comporta da sequencer e da editor per gli eventi MIDI registrati; è, inoltre, possibile gestire i generatori di suoni interni ad Amiga per utilizzarli parallelamente alla strumentazione MIDI. Grazie alla nuova implementazione MPE, di cui parleremo, già tra l'altro presente sul Tiger, è addirittura possibile lavorare con moduli aggiuntivi, secondo la filosofia Music-X, e ampliare quindi le già elevate prestazioni del nostro programma. Tra i moduli già disponibili, vale la pena di ricordare subito Automix, un mixer digitale MIDI, presente sul dischetto del KCS e il famoso Copyist, ora MPE compatibile e quindi molto più versatile nell'interazione con il sequencer.

La struttura del programma

Il KCS è stato progettato per lavorare in tre distinti modi: il Track mode, già presente, in forma meno elaborata sulla precedente versione, l'Open mode e il Song mode. In Track mode, il programma si comporta

Track Mode, schermata Play/Record; questa è la videata che appare subito dopo il caricamento

Schermata del Track Edit, in cui le singole tracce vengono visualizzate sotto forma di eventi MIDI e in cui vengono editate

Schermata Environment, in cui vengono decisi parametri generali, riguardanti il funzionamento dell'intero programma

come un registratore digitale a 48 tracce L'Open mode è un sistema di registrazione più generico che permette di lavorare con concatenazioni di sequenze, fino ad un massimo di 128, un po' come fa il Music-X

Il Song mode serve a concatenare le sequenze e i pattern creati nei due modi precedenti, per realizzare le canzoni finite

Un pezzo musicale può essere creato in uno dei primi due modi (Track o Open) oppure in tutti e due, senza problemi nel passare dall'uno all'altro. Ognuno dei tre modi ha le proprie videate di Play/Record e Edit. Vediamoli, uno per uno, nel dettaglio

Track Mode

La schermata Track Play/Record appare all'inizio del nostro lavoro e si presenta molto compatta E' abbastanza simile, graficamente, a quella presente sulla precedente versione del KCS, ma vi sono state apportate sostanziali modifiche. Lo schermo è diviso in tre sezioni più quella della barra dei menu (vedi figura) la parte dedicata alla visualizzazione delle 48 tracce, nella porzione superiore dello schermo, il pannello dei controlli, nella parte centrale, che contiene la maggior parte dei comandi più usati e, infine, nella parte inferiore dello schermo, la linea dei messaggi, utilizzata per visualizzare l'entrata in funzione di tutti quei controlli che non sono visualizzati nel pannello centrale La maggior parte dei controlli sono raggiungibili sia via mouse che via tastiera del computer (in proposito, viene fornita nella confezione una mascherina di cartone da applicare sulla tastiera, come riferimento sui tasti da usare), alcuni comandi sono attivabili solo da tastiera, altri sono raggiungibili solo tramite menu a finestre, ma per tutti possiamo parlare di "user-friendly", perché sono sempre basati sulla intuitività e sull'immediatezza di utilizzo Il pannello centrale è, a sua volta, diviso in tre parti. a sinistra trovano posti i tasti per l'esclusione dalla registrazione di alcuni controlli (per un risparmio di memoria) quali il Control Change e l'Aftertouch.

Ci sono i tasti per il MIDI Merge e il Rechannelize, quelli per il Mute, il Solo, il Quantize e quello per entrare in Edit Al centro c'è la plancia del registratore vero e proprio, con l'aggiunta dei tasti per la ri-registrazione di una traccia, per la cancellazione dell'ultima e lo slide del tempo Sotto tale plancia c'è il display che visualizza i dati relativi al tempo di registrazione, in minuti e in battute e il metronomo A sinistra ci sono i comandi per il Cue e altri comandi relativi alla gestione delle tracce (Punch.)

Questa descrizione mette in evidenza l'estrema completezza di questa videata che raccoglie in sé tutti i comandi utili Il KCS prevede una impostazione iniziale, detta di default, di alcuni valori, quali, ad esempio, la durata della registrazione, il canale MIDI su cui agisce la funzione Rechannelize, i colori dello schermo e tutte le altre importanti funzioni più usate Questa predisposizione viene caricata direttamente con il programma E' possibile modificarla a piacere grazie alle funzioni Load e Save ENV, che servono, appunto, a salvare e a caricare dei file particolari, nei quali sono contenute le predisposizioni di questo tipo In questo modo, è data all'utente la possibilità di personalizzare il programma a piacimento, senza che questi debba fare la fatica di reimpostare ogni volta i vari parametri

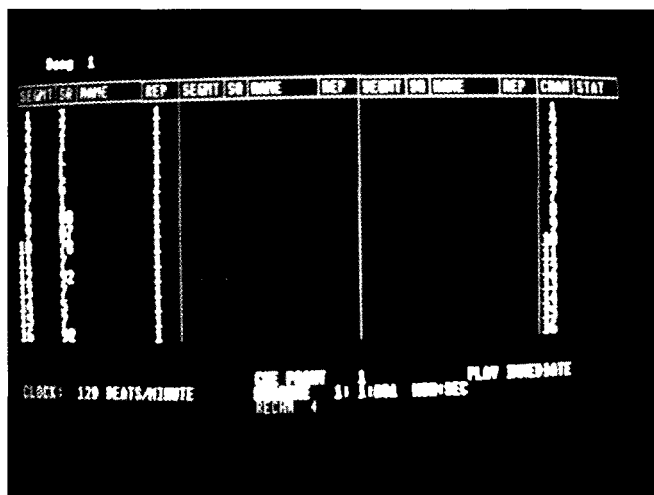
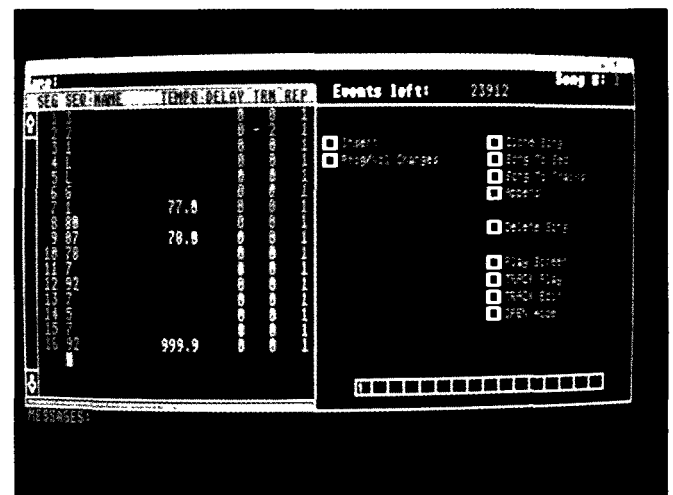
La registrazione può partire in due modi premendo il tasto play oppure suonando una nota sulla tastiera o master keyboard collegata La traccia 1 ha una funzione particolare serve a determinare la lunghezza, in tempo e in battute, della registrazione Si tratta, in pratica, di

una sorta di "traccia pilota", la cui durata determina la durata del brano Questo metodo per determinare la lunghezza della registrazione non è molto usuale nei sequencer MIDI Siamo abituati, infatti, alla solita finestra Requester, in cui selezioniamo il numero delle battute Quando la registrazione è partita procede in modo dinamico per tutte le tracce fino a quando non viene interrotta In altre parole, una volta registrata la prima traccia, il sequencer passa direttamente alla seconda, fino a quando questa non ha raggiunto la durata della prima, per poi passare alla terza e così via

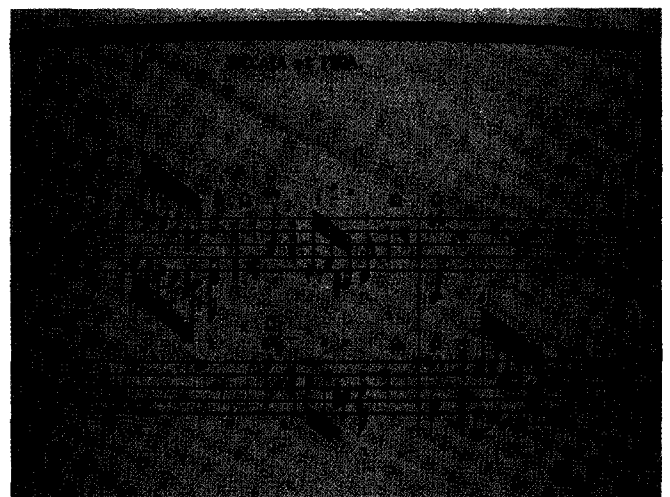
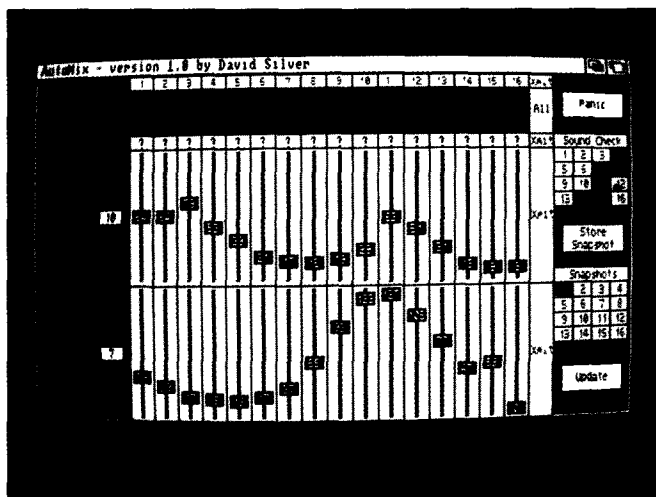
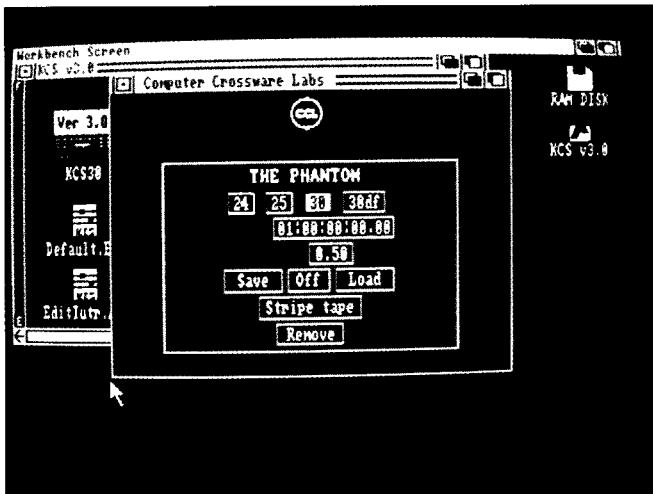
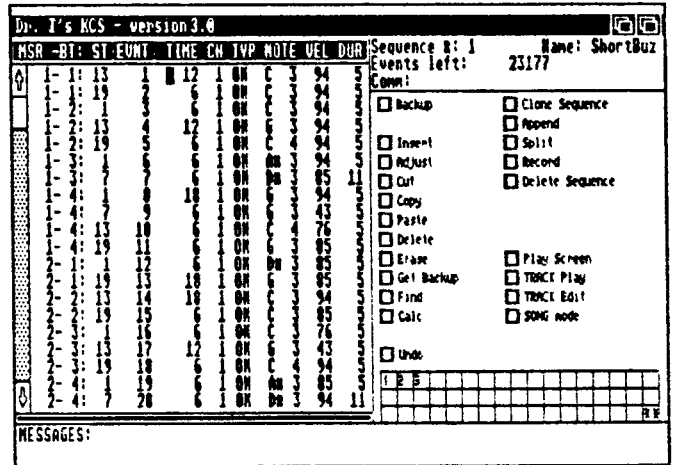
Il nostro consiglio è quello di registrare a vuoto la traccia uno, per un numero di battute superiore a quello previsto dal pezzo, in modo da avere un certo margine e poi, in fase di editing finale del brano, sistemare la durata con la funzione Align, che serve a forzare la durata delle tracce entro un valore stabilito Altra possibilità è quella di entrare in Edit e di impostare il primo evento della traccia uno con la durata del pezzo Solo a prima vista è complicato è vero che la durata viene calcolata in base ai frame (1/120 di beat, per default, modificabile a piacere) e quindi 16 battute sono qualcosa come 15000 frame Tuttavia, accanto alla durata, appare, automaticamente, il numero della battuta e dell'evento corrispondenti ai frame scelti, cosa che facilita il compito notevolmente C'è, inoltre, da dire che, di solito, il numero di battute nelle registrazioni è abbastanza ricorrente (8, 16, 32 battute) Quindi, una volta segnati i numerini corrispondenti a queste battute, la scelta della durata del brano diventa rapida (soprattutto visto che è possibile, come abbiamo detto prima, salvare tali impostazioni in appositi file) Sono davvero molte le funzioni presenti nella videata Track Play è possibile dare nomi alle tracce (6 lettere, per ragioni di spazio), usare il Mute, il Solo, cancellare una traccia, il tutto utilizzando il mouse I menu a finestra che compaiono dall'alto, danno la lista di tutte le operazioni possibili e, quella selezionata tra di esse, è operante tramite il mouse Finalmente, l'utilissima funzione Merge Track è diventata rapida e intuitiva da usare è sufficiente cliccare con il mouse le tracce da fondere e premere Enter, dopodiché il sequencer le metterà tutte nella prima di esse Di rapido accesso anche la funzione Punch che serve, lo ricordiamo, a ri-registrare una parte di una traccia, senza doverla rifare completamente Una funzione davvero innovativa e interessante è quella che mette in collegamento il Track e l'Open Mode E' possibile inviare tutte le tracce registrate in Track Mode ad una sequenza determinata, nell'Open Mode, con una sola operazione Fin qui niente di straordinario Il bello è che si può fare anche l'inverso mandare il contenuto di una sequenza alle tracce Ecco a cosa può servire se per caso dovete trasferire dei dati musicali via MIDI, ad esempio, mandare un brano, costituito da diverse parti, relative a diversi canali MIDI, da un sequencer al KCS, tramite l'interfaccia MIDI, il KCS sarebbe costretto a ricevere tali dati su una sola traccia (a meno che voi non trasmettiate una traccia alla volta, lavoro infausto)

Tale traccia conterrebbe dati per più canali MIDI diversi e un eventuale editing sarebbe impossibile Grazie alle due funzioni sopra descritte, potete mandare ad una sequenza la vostra unica traccia e, da qui, scomporla in tante tracce quanti sono i canali MIDI presenti

Il tutto con sole due cliccate!! Se in questo caso risulta utile questo modo di procedere, per evitare di trasmettere una traccia alla volta da un computer all'altro, diventa indispensabile quando vi trovate a registrare un demo proveniente da una tastiera, in cui tutti i canali vengono mandati contemporaneamente Si

Modo Song, schermo Play Qui vengono visualizzati i segmenti che, presi nell'ordine formano la Song

Modo Song, schermo Edit, dove le singole sequenze vengono concatenate ed editate

tratta davvero di una funzione interessante Novità interessante per gli amanti della registrazione tradizionale è il comando "Send to Fostex R8", che, una volta azionato, rende tale registratore comandabile dal sequencer, in perfetto sincronismo con quest'ultimo Si può fare uno scambio di posizione tra due o più tracce. L'utilità di questo comando, è data da una lentezza intrinseca del sistema MIDI, per cui le ultime tracce, soprattutto quando se ne usano molte, hanno un ritardo leggerissimo rispetto alle prime. Con questo comando si può mettere all'inizio le tracce più "critiche" (per complessità, risposta dello strumento o altro), in modo da ridurre questi problemi di sincronismo La funzione forse più innovativa di questa schermata è il cosiddetto Live Edit Vi permette di cancellare delle note da una traccia o di modificarne la Velocity, mentre la traccia sta suonando, ovvero in tempo reale E' abbastanza interessante perché può capitare, a volte, di registrare una nota di troppo e, poi, non riuscire a trovarla nella schermata di edit (si pensi alle note "sporche")
In questo modo è possibile cancellarla mentre la si ascolta Oppure può capitare di registrare una nota con una velocity troppo elevata la si può facilmente modificare con l'ausilio di pochi tasti Dalla schermata Track Play/Record, premendo il tasto Edit, si passa nella schermata Track Mode Edit (che, come vedremo, è assai simile alla schermata Open Mode Edit) Qui vi è data la possibilità di operare singole o globali modifiche su parametri quali la frequenza, la durata, la velocity, il canale MIDI e il tempo, per ogni evento MIDI registrato. Lo schermo di Edit è diviso in quattro sezioni nella parte alta a sinistra c'è la lista degli eventi MIDI registrati, a fianco trovano posto alcune delle opzioni di editing, la finestra di selezione delle tracce, sotto alle opzioni e la linea dei messaggi in fondo La lista degli eventi è divisa, a sua volta, in otto colonne verticali, in ciascuna di esse vengono visualizzati i seguenti dati il numero della misura, del beat e dello step in cui ci troviamo, il numero dell'evento, la durata di questo evento rispetto a quello precedente, il canale MIDI, il tipo di evento considerato, la nota, la velocity e la durata assoluta della nota
Una volta presa l'abitudine, leggendo queste otto informazioni per ogni evento registrato, è possibile capire e modificare ogni informazione che lo riguarda Lo scroll degli eventi si è dimostrato molto rapido e quindi, anche una traccia molto lunga può essere percorsa in tempi accettabili Nella finestra degli eventi, c'è un cursore, posizionato nel punto in cui verranno inseriti gli eventi La posizione del cursore è molto importante in fase di ascolto, perché è possibile, in questa schermata, premendo il tasto destro del mouse, ascoltare la traccia che si sta editando L'ascolto parte dal punto in cui è posizionato il cursore In basso a destra si trova, in fase di ascolto, il contatore delle battute La finestra delle tracce serve a selezionare la traccia che si vuole editare cliccando con il cursore del mouse in uno dei 48 quadratini della griglia, le informazioni contenute nella traccia selezionata verranno visualizzate nella finestra degli eventi Nella parte alta a sinistra trovano posto, come ho detto, alcune delle opzioni di editing, cliccabili direttamente da mouse, tra le più comuni Altre opzioni sono disponibili nei menu a finestra in alto. Singolare notare che è presente una "lavagnetta" in cui si possono scrivere delle linee di testo e di appunti, al posto dei soliti foglietti volanti
Una veduta complessiva del Track Mode mette in luce come, pur rimanendo un sequencer tradizionale, il KCS 3 0 offra alcune innovative e interessanti possibilità di impiego che, certamente, faciliteranno e velocizzeranno l'utilizzo da parte dell'utente

Automix, il primo modulo MPE disponibile per il KCS 3 0, fornito nel dischetto del programma

Il Copyist, il programma Dr T's per la gestione degli spartiti, ora MPE-compatibile

Phantom, il programma per il sincronismo SMPTE, fornito nel dischetto del KCS

La schermata del Track Editing

Open Mode

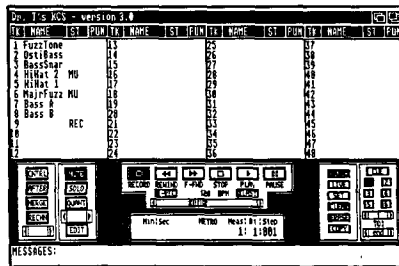
Vediamo ora in cosa consiste la vera novità. Si tratta, come accennato, di un metodo di registrare sganciato dal concetto di traccia. Si possono definire fino a 128 sequenze indipendenti, qualunque numero delle quali può essere suonato simultaneamente. Alcune sequenze possono includere comandi di Start/Stop per altre sequenze e così via, concatenando in questo modo le diverse sequenze a formare una song. La schermata di Edit è molto simile a quella di Edit presente in Track Mode, solo che qui, invece delle singole tracce, vengono trattate intere sequenze. Il primo vantaggio riscontrabile lavorando con le sequenze, anziché con le tracce, è certamente quello che si può prescindere dal Song Mode, in cui si concatenano le tracce, e, volendo, anche le sequenze, per creare delle song. Infatti, mentre non è possibile, in Track Mode, concatenare singole tracce, in modo che vengano eseguite in tempi diversi, l'Open Mode offre la possibilità di generare sequenze di controllo, il cui compito è proprio questo: dirigere l'orchestra delle altre sequenze, dicendo loro quando suonare e quando smettere. In effetti, entrambi i modi di registrazione si sono rivelati validi: in Open Mode sono infatti disponibili tutte le funzioni descritte per il Track Mode. Si tratta quindi di comodità personale nell'utilizzo. Da parte nostra, abbiamo proceduto così: ci siamo serviti del più tradizionale sequencer del Track Mode per creare i singoli Pattern, formati da più tracce. Poi abbiamo copiato i Pattern su singole sequenze e, infine, abbiamo legato assieme queste ultime, così da creare il pezzo completo. Per quanto riguarda la fase di Editing, consigliamo di lavorare sulle singole tracce, anziché sulle sequenze, perché può capitare che su una sequenza siano registrate informazioni destinate a più canali MIDI diversi. In ogni caso, l'Open Mode rappresenta una grossa innovazione per quanto riguarda i sequencer disponibili per Amiga e siamo felici che proprio il KCS 3.0 abbia introdotto, nel mondo Amiga, questo nuovo modo di fare musica su sequencer.

Song Mode

Due parole anche sulla parte Song. Assolve, in pratica, la funzione appena descritta nell'Open Mode: serve a concatenare singole sequenze per creare una Song. Il vantaggio del Song Mode è che questa operazione risulta assai più rapida e semplice da eseguire. Ogni Song può contenere fino a 100 segmenti, ciascuno dei quali può essere ripetuto quante volte si vuole. Si può, inoltre, specificare, per ciascun segmento, dei valori per il tempo, per il delay e per la trasposizione; è possibile, inoltre, assegnare ad ogni canale MIDI un numero di program change. Il tutto per ogni segmento che andremo a legare alla catena: non mi sembra poco. Inutile aggiungere che ad ogni song che create potete assegnare un nome, lungo fino a sedici caratteri.

Le altre novità

Vediamo ora le caratteristiche più generali del programma, che contribuiscono a farne un pacchetto integrato professionale. Novità, per il KCS, è la finestra per la gestione dei generatori di suoni di Amiga. Abbiamo già rilevato la tendenza a sfruttare le capacità sonore del nostro computer già in altri programmi, primo fra tutti il Deluxe Music Contruction Set. Il primo programma professionale a farne uso è stato il Music-X, in cui è addirittura possibile editare la forma d'onda dei suoni disponibili. Il KCS permette di caricare in memoria fino a 16 suoni alla volta, di cui, ovviamente, se ne possono usare, nello stesso momento, solo quattro. Ai suoni viene associato un canale di riconosci-

La schermata del sequencer nel Track Mode

mento, simile, per concetto, al canale MIDI. Di ogni suono si può stabilire il registro (Range) e la trasposizione. Sebbene i suoni di Amiga non siano paragonabili al livello di professionalità a cui si rivolge il sequencer, è comunque utile poterli sfruttare, magari anche solo per modificare il tic-tac del metronomo (il cui suono è, tra l'altro, generabile anche via MIDI), oppure per generare effetti speciali per i quali si dovrebbe sacrificare un prezioso canale MIDI. E' ovviamente presente la funzione di registrazione Step-by-Step. Lo schermo è così suddiviso: ci sono, al centro, quattro colonne di numeri e lettere, che specificano tutta la gamma di note, quindici diverse durate, quindici diversi tempi di inizio nota e sei diverse velocity. Le selezioni possono essere fatte sia dalla tastiera del computer e dal mouse, che dalla tastiera del vostro sintetizzatore o master keyboard, da cui sarà automaticamente specificata la velocity delle singole note. La registrazione Step-time può essere effettuata sia dal Track Mode che dall'Open Mode, selezionando la funzione dai menu a finestra. All'inizio della registrazione, il programma chiede da quale battuta si vuole partire. La registrazione è molto semplice perché, una volta selezionati i parametri, è sufficiente inserire, una ad una, le note. Per velocizzare ulteriormente le cose, si è pensato di inserire, in fondo allo schermo, una finestrella per cambiare rapidamente la durata delle note. L'ultima novità è l'MPE, Multi Program Environment. Si tratta di un nuovo metodo per permettere a diversi programmi di accedere direttamente agli stessi dati. La differenza sostanziale con il Multitasking, consiste nel fatto che, nel passaggio da un programma all'altro, non è necessario fare copie, su disco, dei dati, da travasare al software successivo. Questo si traduce in un risparmio notevole di memoria perché i dati vengono caricati una sola volta per tutti i programmi presenti e in un notevole risparmio di tempo, necessario per svolgere tali operazioni. Così, si parte da un programma principale, quale può essere, nel nostro caso, il KCS 3.0, e da qui si caricano gli altri, che vengono detti moduli, dato che dipendono dal primo. Il menu MPE, presente nella barra in alto del KCS, si allunga mano a mano che vengono caricati moduli. E' possibile gestire, contemporaneamente, fino a diciannove moduli sempre che, specificano i tecnici Dr. T's, abbiate abbastanza memoria e siano disponibili abbastanza moduli. Per ora sono stati sviluppati solo tre moduli: il primo è allegato al KCS ed è registrato sullo stesso dischetto. Si tratta di Automix, un mixer digitale a 16 piste, una per ogni canale MIDI, in cui si può modificare il volume dei singoli canali e il loro Pan-Pot. E' possibile fare delle "fotografie" delle singole situazioni, che possono così essere immagazzinate. Il mixer funziona in tempo reale con il sequencer ed è così possibile variare i vari livelli mentre si ascolta il brano. Un secondo modulo è presente nella confezione di Tiger, altro programma della ditta americana, e il suo nome è QuickScore, un software molto carino che permette di visualizzare, sotto forma di spartito, la musica registrata con il sequencer, e di stamparla su carta. Tale modulo non consente però un editing di tale spartito. A tal proposito è recentissima l'uscita del Copyist, il programma di stampa musicale, in versione MPE-compatibile, che può quindi essere utilizzato come modulo del sequencer. Per gli amanti del sincronismo S.M.P.T.E. è presente, sul dischetto del KCS il programma Phantom, che funge da sincronizzatore, in tale standard, durante il passaggio di informazioni.

Conclusioni

Non rimane molto da dire, visto che, durante la descrizione del programma, abbiamo approfondito tutti gli aspetti più interessanti. Il fatto più notevole è che finalmente la Dr.T's si presenta con un pacchetto integrato in grado di competere con gli altri software professionali. Ciò che è ancora più interessante è che il KCS 3.0 non si presenta come un pacchetto statico, fine a se stesso e, quindi, soggetto ad obsolescenza. La possibilità, infatti, di utilizzare i moduli MPE ne incrementa i modi d'uso e la durata nel tempo. Senza contare che sarà presto possibile fare affidamento su un certo numero di programmi-moduli dedicati all'editing delle voci dei vari sintetizzatori ed expander, cosa che pareggerà i conti con Music-X che, come i nostri lettori ricorderanno, possiede già tale facility. Si tratta, insomma, di un programma in espansione, pronto ad accogliere le novità del mondo MIDI. Sottolineiamo che è in commercio anche il programma Level II, sempre Dr. T's, che, al KCS 3.0 appena descritto, affianca la capacità di generare improvvisazioni musicali su temi dati. Lascio alla vostra fantasia il compito di stabilire cosa si potrà fare con la linea Dr. T's. Il manuale del KCS si snoda attraverso una serie di tutorial, uno per ogni sezione del programma che mettono in grado coloro che, magari per pigrizia, non vogliono leggere tutto il manuale, di utilizzare il programma nelle sue linee essenziali. Seguono poi capitoli dedicati all'approfondimento dei singoli aspetti delle varie parti del software, nonché, per finire, un capitolo dedicato agli utenti avanzati, in cui vengono svelati trucchi e segreti per ottenere il massimo da questo programma. ▲

Ringraziamo la ditta Soundware, che ha gentilmente messo a disposizione il software per le prove.

I File e il Linguaggio C

Gianni Biagini

Questo mese parliamo, come avrete intuito dal titolo, di file.
Cosa essi siano, è abbastanza noto, molti definiscono il file come un blocco di dati "messo" da qualche parte, in genere su un dischetto.
La definizione in sé stessa non è scorretta, però molto approssimativa, soprattutto in relazione alle nuove tecniche di programmazione.
Il concetto di file è molto comune nella vita quotidiana, supponiamo di mettere i nostri documenti in una cartelletta, abbiamo già creato, senza rendercene conto, un file, così è anche per il portafoglio e per il classico "dimenticatoio", un file particolare che informaticamente chiameremo NIL:.
Il file informatico è un po' più restrittivo degli esempi citati, la limitazione maggiore, è che tutti gli elementi in esso presenti devono essere di tipo compatibile, cosa comporta questo?
Innanzitutto, nel portafoglio posso mettere banconote ma anche documenti, quindi esiste una tipologia diversa.
Tolta questa restrizione (peraltro i più moderni database ipermediali permettono di accedere a dati di tipo diverso e con modalità di ricerca potentissime) il nostro file campione può vantare una forte duttilità, innanzitutto un file non deve essere necessariamente composto di soli caratteri o di soli numeri, possiamo definirlo come file di strutture, per esempio, il che, in pratica, si riduce ad uno schedario come quello dei medici dove ad ogni paziente sono associati una serie di dati relativi alle sue condizioni fisiche.
Se a questo punto vi siete fatti un'idea del file di memorizzazione, aggiungete un alro tipo di file molto importante: le periferiche.
L'interfaccia utente CLI, quella che vediamo ogni volta che usiamo AmigaDOS, è vista come un file di Input/Output, la stampante come un file di solo Output e la tastiera come file di solo Input.
Questa concezione di intendere come file ogni meccanismo di input/output, è un concetto derivato dal sistema operativo UNIX, dove riveste un'importanza fondamentale; il linguaggio C d'altronde, essendo un prodotto tipico del mondo Unix, ne assume molte delle caratteristiche fondamentali, abbiamo infatti tre file di grande importanza, essi sono l'input standard, l'output standard e l'output degli errori. Poiché non vige una legge che impone agli Input di un programma di provenire dalla tastiera, è possibile "redirigere" l'input, ossia accettare dati da un dispositivo diverso (sia esso un modem, piuttosto che una penna ottica, piuttosto che un file). Lo stesso, ovviamente, vale per la redirezione dell'output e per la gestione degli errori (potrebbe rivelarsi utile avere un file traccia di tutti i problemi riscontrati).
Addentriamoci più a fondo nei file che ci interessano in questo momento, quelli che servono a memorizzare i dati, in particolar modo i caratteri.
Il linguaggio C, molti di voi lo sapranno, deriva da uno strano parente, il BCPL, da cui ha ereditato una struttura importantissima: il File Header.

```
struct FileHandle  {
  struct Msg      *fh_Link;
  struct MsgPort     *fh_Port;
  struct MsgPort     *fh_Type;
  LONG            fh_Buf,
                  fh_Pos,
                  fh_End,
                  fh_Funcs,
                  fh_Func2,
                  fh_Func3,
                  fh_Args,
                  fh_Arg2;
```

Il File Handle non riveste grossa importanza per noi in questo momento, viene utilizzato per colloquiare con quella struttura interna del sistema operativo chiamata nocciolo (il Kernel, per l'appunto).
La struttura elencata ci serve per capire che ogni file consta di una serie di parameri, essi sono legati alla sua ubicazione sul disco, a buffer per la scrittura e la lettura e ad altre procedure specifiche.
Certo in Basic o in Pascal la gestione è più semplice, ma sicuramente meno potente.
Innanzitutto, vediamo come creare o aprire un file esistente, impresa per altro molto semplice visto che ci sono una serie di funzioni di libreria per questo sviluppate.
Chiaramente il verbo OPEN, che in inglese significa "apri", è il più indicato per accedere ad un file, sia esso un archivio o una unità logica, non ci meraviglierà, quindi, una sintassi del genere:

```
file_handle=Open(nome_file,modo)
```

Notiamo innanzitutto che nome_file deve essere una stringa, sia essa una variabile o una costante, per aprire il file dal classico nome pippo.doc basterà scrivere:

```
file_handle=Open("pippo.doc",modo);
```

La variabile file_handle, è una struttura specifica, essa rappresenta il FileHandle del BCPL che abbiamo visto precedentemente, quando accediamo ad un file, i vari campi della struttura vengono aggiornati di conseguenza.
Interessante è notare l'ultima variabile che specifica le modalità di accesso al file, essa può assumere due diversi valori, ricordatevi di scriverli sempre in maiuscolo:

```
modo = MODE_OLDFILE

modo = MODE_NEWFILE
```

certamente queste due modalità vi ricorderanno il RESET e il REWRITE del Pascal, sono in effetti molto simili.
Aperto il nostro file e ultimato il nostro programma, non resta altro che chiuderlo, ricordatevi sempre di farlo, è molto importante, innanzitutto per risparmiare memoria (e ogni utente sa quanto essa è utile) e, in secondo luogo, per rendere quel file accessibile anche ad altri utenti o ad altri processi (soprattutto in relazione al fatto che Amiga è multitasking).
Il comando in questione, neanche a dirlo, è il contrario di apri, chiudi o, meglio ancora, l'equivalente in inglese, CLOSE.
I parametri prima specificati non ci interessano più, nome e modo operativo sono ora inutili poiché è stato inizializzato il file_handle che racchiude in se tutte le caratteristiche fondamentali per la gestione dello stesso; la procedura di chiusura quindi, si riduce a:

```
Close(file_handle);
```

Prima di proseguire con lo studio delle funzioni di lettura e di scrittura, ripassiamo un concetto che in informatica riveste una grossa importanza, la bufferizzazione.
Quasi tutti avranno un'idea, più o meno completa, di cosa sia un buffer, si tratta di un tampone.
Facciamo un semplice esempio di buffer, come al solito ci atterremo alla tradizione degli esempi delle rubriche di "programmazione facile", parleremo in termini enogastronomici, come il buon Gianni Brera.
Pensiamo ad una umida e polverosa cantina, l'ideale per far invecchiare il vino, ci arriva una botte di "quello buono", la nostra prima preoccupazione è quella di imbottigliarlo.
Ci sono in tutto tre elementi, quello di partenza ossia la botte, il tramite cioè l'imbuto e, infine, la bottiglia l'elemento di arrivo.
Noi ci occuperemo principalmente dell'imbuto. La trasposizione non ci deve meravigliare più di tanto, non a caso si parla spesso, in informatica, di collo di bottiglia.
Iniziamo a versare il vino, l'imbuto è infilato nella bottiglia, dopo poco tempo esso si riempie perché la velocità di uscita dalla botte è maggiore di quella in entrata della bottiglia, l'imbuto d'altronde, grazie alla sua forma, permette di vedere quando il prezioso liquido straripeerebbe, noi fermiamo il getto e attendiamo che l'imbuto sia vuoto.
L'imbuto rappresenta degnamente un buffer, il vino esce troppo velocemente dalla botte, cosicché ci serve un tampone alla fuoriuscita.
Vogliamo rendere quanto detto molto più simile al linguaggio informatico?
Quando si riempie l'imbuto siamo nella condizione di Buffer Empty, il nostro Amiga ci può segnalare il tutto con un alert, che suonerebbe in questo modo:

```
Buffer Empty, click left mouse button to resume
```

Basterebbe fare click sul gadget di ok per riiniziare la mescita.
Un buffer, quindi, è un qualche cosa (non necessariamente una memoria perché abbiamo visto che l'imbuto lo impersona benissimo) che deve fare da tramite tra due meccanismi con una diversa velocità operativa, nel nostro caso creeremo una zona di memoria atta a contenere i dati in lettura dal drive che dovranno essere processati attraverso il nostro programma.
Forse qualcuno starà pensando che potremmo leggere i dati carattere per carattere, così da non aver problemi di disparità tra le velocità dei meccanismi solo che, ve lo assicuro, oltre ad essere dannoso per il drive (che sarebbe costretto a girare ininterrottamente), rallenterebbe in maniera inaccettabile il corso del programma.
A riprova di quanto detto, basti pensare alla voce informatica Buffered I/O ossia Input Output bufferizzato, è universalmente utilizzata.
Stabilito che i buffer non servono solo per imbottigliare il vino, vediamo come utilizzarli per accedere a dei dati da un disco.
La funzione di lettura, al proposito, ci riserva il primo esempio di quanto detto, il suo nome, è:

```
byte_letti=Read
(file_handle,buffer,ampiezza_buffer)
```

byte_letti conterrà, alla fine della lettura, il numero di byte letti (deve essere, quindi, una variabile del tipo standard intero), file_handle è proprio quello associato al file al momento dell'apertura, buffer è una zona di memoria tampone e, infine, ampiezza_buffer rappresenta il numero di locazioni occupabili nel buffer (anche questa variabile deve essere di tipo intero).
Altro fattore da tenere in considerazione, parte della filosofia del linguaggio C, comprende la gestione degli errori, che è sempre lasciata al programmatore, in questo caso fate attenzione che se il valore di byte_letti è uguale a 0 vuol dire che abbiamo raggiunto la fine del file, e, infine, se il valore è uguale a -1 si è verificato un errore.
Naturalmente la nostra Read è compagna dell'altra funzione:

```
byte_letti=Write
(file_handle,buffer,ampiezza_buffer)
```

Non occorre ricordare che -1 rappresenta una condizione di errore e che 0 non può rappresentare la fine del file (logico poiché lo stiamo

(segue a pag. 71)

Amiga in rete

Sebastiano Vigna

Questo mese vorrei raccontarvi dell'esperienza fatta con PARNET, un programma che dovrebbe essere molto più diffuso di quanto non sia effettivamente. Come sempre spero che leggere gli errori altrui possa distogliervi dal ripeterli (ma su questa falsariga non dovrebbero esserci più guerre, e quindi non ci conto più di tanto...).
PARNET è il risultato di una collaborazione tra Matt Dillon (che conoscete tutti benissimo) e la Software Distillery, in particolare Douglas Walker e John Toebes (sì, quelli che curano il SAS/C). Questi ultimi, basandosi sulla parnet.device di Matt Dillon, che permette di vedere la porta parallela di Amiga come un connettore di rete, hanno scritto il software di gestione per una semplice rete Amiga <-> Amiga. In pratica, dati due Amiga e un cavo opportuno (la cui costruzione è descritta nella documentazione) è possibile collegarli in rete a circa 28K al secondo. "Collegarli in rete" significa in questo caso che da ciascuno dei due Amiga è possibile accedere al filesystem dell'altro. Se, ad esempio, sul primo Amiga esiste una partizione di hard disk chiamata DH0:, sul secondo Amiga è possibile scrivere "list NET:dh0" e leggere la relativa directory. Tutte le operazioni che passano per il filesystem sono permesse (copie, rename, ecc.). La cosa più interessante è, ovviamente, la copia remota: nel caso specifichiate una sorgente sull'altro computer e una destinazione sul vostro, PARNET trasferirà trasparentemente il file attraverso il cavo di comunicazione. In pratica, i due filesystem vengono identificati a tutti gli effetti.
Le applicazioni di PARNET sono numerosissime: dalla copia diretta da hard disk a hard disk su computer diversi, all'uso come piccola rete. Il tutto gratis, dato che il programma è freeware. Ad esempio, per le macchine MS-DOS esiste un celebre programma commerciale di comunicazione, il LapLink, che collega due macchine attraverso la porta seriale e permette di scambiare file tra di esse.
Vengono visualizzate due finestre, la prima delle quali mostra il proprio filesystem, mentre la seconda mostra quello remoto, ed è possibile copiare da una all'altra. PARNET va molto più in là: mettendo in collegamento i due filesystem, è possibile utilizzare qualunque file utility (DiskMaster, MegaD, DirOpus, SID...) per fare la stessa cosa. Non è neppure necessario utilizzare la stessa utility sui due Amiga. E il trasferimento, avvenendo attraverso la porta parallela, è molto più veloce. Il merito, oltre che ovviamente degli autori, è della strutturazione spinta del sistema operativo di Amiga, che isola in maniera ferrea le componenti del sistema, secondo la migliore pratica di astrazione, permettendo la loro sostituzione o modifica con la massima affidabilità.
Sfortunatamente, può capitare di incorrere in qualche inconveniente... come nel mio caso. Dato che avevo una voglia matta di provare PARNET, comprai il necessario per costruire il cavo e mi misi al lavoro. C'erano una dozzina di cavi da collegare, più le masse. Le masse? Perché perdere tempo?
Colleghiamone una sola, porterà tutti i segnali insieme.
Mai sia! Dopo aver fatto un paio di directory remote, PARNET smise di funzionare, e per quanto tentassi, non ci fu modo di riattivarla. Dopo diverse ore di test, scoprii che la scheda di espansione RAM del secondo Amiga era andata a farsi friggere, e probabilmente uno dei due CIA aveva subito la stessa sorte. Morale: collegate tutte le masse (c'era pure scritto nelle istruzioni...). Qualcuno mi ha spiegato perché, ma non ho capito pressoché nulla.
Un altro collaboratore di Amiga Magazine ha tentato il mio stesso esperimento, con la variante non indifferente di fabbricare il cavo giusto, e ora può eseguire backup da computer a computer a velocità paurosa. Non so se avete fatto il conto, ma 28K al secondo sono 229376 bps, che per una soluzione non dedicata (non c'è alcun tipo di hardware aggiuntivo a parte il cavo) non è proprio male.
PARNET è, quindi, l'ennesimo prodotto di altissima qualità che esce dalla Software Distillety. John e Doug stanno attualmente lavorando ad altri miglioramenti per farne una vera e propria rete low-cost. Ne vedremo delle belle...
Sempre sul versante comunicazioni, vorrei spendere due parole sul modem interno della Applied Enginnering di cui avrete probabilmente visto la pubblicità su qualche rivista americana. Dopo alcuni mesi di onorabile servizio, ha cominciato a dare i numeri. All'accessione emetteva mostruosi suoni tramite l'altoparlantino, e non rispondeva ad alcun comando. Ho mandato il tutto alla Applied Engineering, che dà una garanzia di 5 anni, e molto gen-

(segue a pag. 77)

Disegnamo con Amiga

Gianni Biagini

Nei numeri precedenti, abbiamo studiato alcune delle funzioni grafiche fondamentali, ora siamo in grado di rappresentare immagini complesse a piacere. La sola teoria, però, spesso non è sufficiente, questa la ragione per cui ci accingiamo ora a mettere in pratica quanto precedentemente visto. Qualcuno di voi forse storcerà il naso sapendo che ci occuperemo di rappresentazioni di funzioni matematiche, la macchina che però possedete, non deve essere vista solo ed esclusivamente come un bel gioco, ha delle potenzialità di calcolo di tutto rispetto. Non ci occuperemo a fondo di algoritmi matematici, che appesantirebbero notevolmente la scorrevolezza del presente articolo, tuttavia attraverso delle rappresentazioni vedremo come creare delle strutture che potranno essere utili in qualsiasi contesto. Detto questo, inoltriamoci nello stupendo labirinto delle funzioni trigonometriche. Piccolo ripasso: esiste un settore della matematica analitica chiamato trigonometria, esso comprende alcune funzioni che sicuramente avrete sentito nominare: seno, coseno e tangente. Il seno rappresenta il rapporto che vige tra la proiezione di un punto della circonferenza sull'asse orizzontale e la misura del raggio, è bene sapere che dà come risultato un numero compreso tra 0 e 1. Lo stesso dicasi del coseno, solo che abbiamo a che fare con la proiezione sull'asse verticale. Adesso staranno torcendo il naso i cultori della matematica, certo la spiegazione è un po' semplicistica ma sufficiente per il nostro caso.
Per non complicarci troppo la vita, non studieremo la tangente, ci occuperemo solo di rappresentarla graficamente, sappiate, comunque, che per un angolo sotteso uguale a pigreca mezzi (novanta gradi) essa diverge, acquista cioè un valore infinito. Altro concetto che incontreremo, è quello di asintoto, proprio nel caso della tangente ne abbiamo un chiaro esempio. Un asintoto non è altro che un valore verso il quale una funzione si avvicina indefinitamente senza però riuscire mai a raggiungerlo. In seguito tracceremo una linea che lo rappresenti graficamente. Scopo di tutto, come avrete capito, è il prendere dimistichezza con le funzioni grafiche. Il nostro breve programma comincia con la definizione di una costante fondamentale, il pi greco. Esso rappresenta il rapporto che vige tra la lunghezza della circonferenza e il suo raggio. Il BASIC come sapete non permette di inizializzare costanti, noi assegneremo alla variable PI un valore che rimarrà inalterato: 3,14159.

PI = 3,14159

Onde differenziare la finestra di output dalla finestra sulla quale vogliamo lavorare, creeremo due nuove strutture, uno screen e una finestra.

```
SCREEN 2,640,200,3,2
```

Abbiamo così creato uno schermo delle dimensioni 640x200 con otto colori in alta risoluzione non interlacciata. Da cosa deduco tutto questo? Dal fatto che il numero di piani dello schermo è uguale a tre e, quindi, avendo 3 bit plane, ho otto possibili colori utilizzabili.
La risoluzione è data, invece, dall'ultimo parametro, come da tabella:

Modo	Tipo di Screen
1	bassa risoluzione non interlacciato
2	alta risoluzione non interlacciato
3	bassa risoluzione interlacciato
4	alta risoluzione interlacciato

Creato lo schermo, passiamo a definire la finestra nella quale visualizzeremo i grafici, la vogliamo della grandezza dello schermo e non ci interessano particolari gadget come, ad esempio, quello di profondità o di chiusura.

```
WINDOW 3,"Studio di Funzioni",,0,2
```

La nostra finestra sarà identificata dal numero 3, avrà per titolo "Studio di Funzioni", coprirà tutto lo schermo (non abbiamo specificato le coordinate quindi per default tutto lo schermo viene coperto), non ha gadget i quali sono dati dalla somma dei valori della seguente tabella:

1	ridimensionamento
2	spostamento
4	profondità
8	chiusura
16	memorizzazione del contenuto

Se volessi avere una finestra chiudibile, spostabile e ridimensionabile, dovrei specificare come parametro 11 ossia 1 + 2 + 8 = 11.
L'ultimo parametro che ci rimane rappresenta lo screen al quale la

finestra è associata, in questo caso 2 visto che lo abbiamo appena creato. Per rendere l'effetto grafico maggiore, vogliamo uno schermo nero come la notte, agiamo allora attraverso la modifica della palette sul colore 0, quello dello sfondo.
Non ricordate come si cambia il valore della palette? Basta inserire i valori di saturazione come parametri alla funzione:

```
PALETTE  numero_colore,  rosso,
verde, blu
```

per il nero questi valori sono tutti nulli, quindi, il nostro programma conterrà il seguente comando:

```
PALETTE 0,0,0,0
```

Da questo momento in poi inizia la routine di visualizzazione delle curve, ci servono due variabili contatore, che rappresenta il valore dell'ascissa della funzione e che assume valori incrementati di uno ad ogni passaggio del ciclo e n che va da 0 a pi greco con un passo di 0,005.

```
FOR N = 0 TO PI STEP 0.005
```

notiamo che il numero decimale è stato espresso con il punto di separazione al posto della virgola, come vuole lo standard americano.

```
PSET (M,100+30*SIN(N)),2
```

Qui stiamo creando un punto, la sua ordinata viene sommata a 100 al fine di avere la visualizzazione nel mezzo dello schermo, il valore SIN(N) è stato, inoltre, moltiplicato per trenta in modo che la curva fosse visibile; ultima particolarità, il valore 2 è quello del colore associato al punto.

```
PSET (M,100),1
```

I punti che vengono tracciati dalla funzione di cui sopra, si trovano tutti allo stesso valore di Y che, per motivi estetici, abbiamo voluto al centro dello schermo, ciò che deriva è un asse di simmetria della curva.

```
PSET (M,100+30*COS(N))
```

Come per la sinusoide, solo che la curva è in questo caso una cosinusoide (rappresenta la funzione coseno).

Pensiamo ora a tracciare la curva della tangente, qui torna utile il concetto di asintoto in quanto il punto in cui la tangente diverge (va all'infinito) viene visualizzato in due modi, attraverso una circonferenza che chiarifica per quale valore di X questo succede e mediante la rappresentazione grafica dell'asintoto stesso. Come abbiamo detto, circonferenza e linea devono essere rappresentati solo se la tangente diverge (nel nostro caso se esce dallo schermo) La condizione di avere un valore maggiore di 100 è spiegata dal fatto che il nostro schermo ha un'altezza in pixel uguale a 200, se, quindi, la tangente sorpassa tale valore "esce" dallo schermo.

```
IF TAN(N)> 100 THEN
   CIRCLE((M,100),25,4
   'è una circonferenza di raggio
    25 e colore 4
   LINE (m,1)-(m,199),1
   'una linea verticale che indica
   l'asintoto della curva
END IF
```

Tracciamo, quindi, il punto della curva della tangente:

```
PSET (M,100+TAN(N)),5
```

chiudiamo finalmente il nostro ciclo:

```
NEXT N
```

Prima di chiudere la finestra e lo screen creati attendiamo che l'utente veda il risultato ottenuto, facciamo perdere un po' di tempo al nostro Amiga:

```
FOR PAUSA = 1 TO 10000: NEXT PAUSA
```

Armiamoci di senso civico e raccogliamo i resti di quanto fatto, chiudiamo tutto:

```
WINDOW CLOSE 3
SCREEN CLOSE 2
```

come ogni programma che si rispetti anche il nostro ha una fine:

```
END
```

N.B. Per un banale errore durante la fase di duplicazione, il programma che accompagna questo articolo è stato registrato anche sul disco del numero precedente. Ci scusiamo con tutti i nostri lettori. ▲

MAGAZINE AMIGA Next...
..nel prossimo numero

▲ **Speciale Hard Disk**
Gli Hard Disk e Amiga 500

▲ **Amiga World Expo**
Tutte le novità della fiera di New York

▲ **Home Video Kit**
Un pacchetto completo per i principianti di DTV

▲ **The Art Department Professional**
La potenza dell'elaborazione grafica

▲ **Podscat PT 3030**
Una tavoletta grafica dalle ottime qualità

▲ **ECTS '91**
Tutte le novità dei videogames fino a Natale

▲ **Vista**
Il magnifico generatore di paesaggi tridimensionali

▲ **Distant Suns**
L'Astronomia con Amiga

Appuntamento in edicola

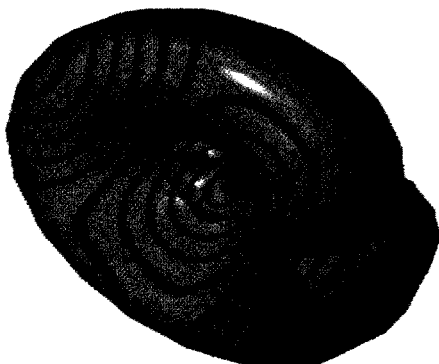
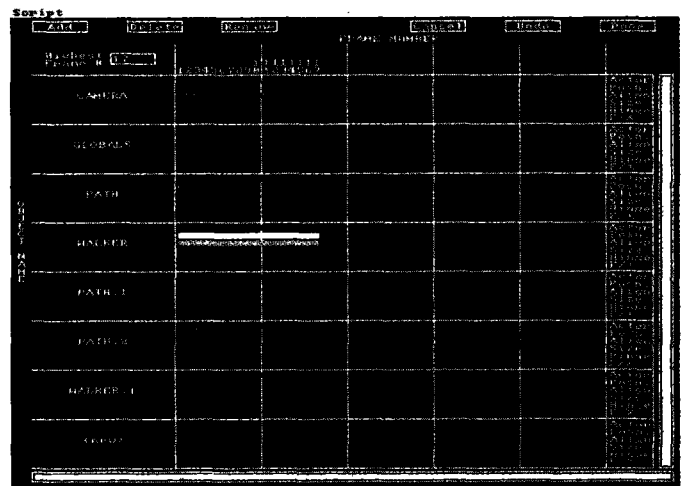
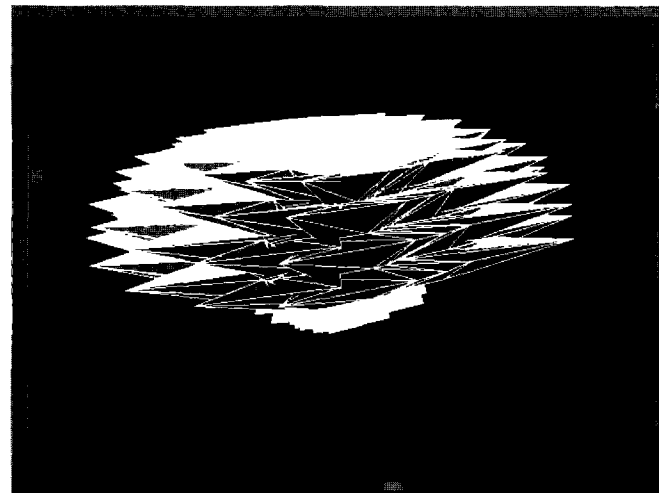
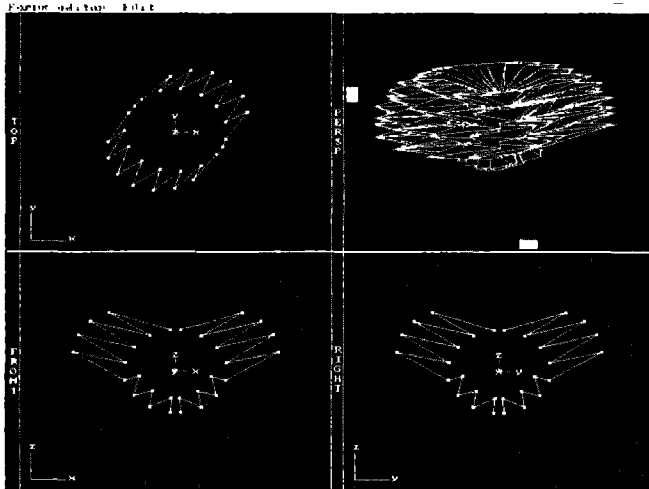
Imagine

Diego Montefusco

Ecco l'atteso seguito del best seller Turbo Silver, primo esponente della cosiddetta terza generazione di programmi grafici per Amiga (insieme ai non ancora disponibili Animation: Journeyman e Lightwave).

La confezione

Per essere un pacchetto da quasi mezzo milione, il packaging non è davvero all'altezza: una anonima scatola di cartone bianco, avvolta in un sottile foglio (recante immagini, nome e caratteristiche del pacchetto) che, per di più, una volta sfilato è difficile rimettere senza rovinare. Una volta aperta la scatola, seconda delusione: un solo dischetto, due manuali striminziti formato A3, su cui ci soffermeremo poi, un addendum, la garanzia e basta. Anzi no! Completa il tutto un etto circa di "chipster" di polistirolo per non rovinare lo scarso materiale che altrimenti navigherebbe nello scatolone.Senza nulla voler togliere alla qualità del software, che è poi quello che conta, quasi, di più, alla Impulse potevano cercare di rendere più professionale il tutto, come ha fatto, per esempio, Commodore col suo, eccellente, AmigaVision. Anche se il software risiede tutto su un unico disco (tanto più che questo è compattato) è ormai prassi comune dotare il pacchetto di qualche dischetto con esempi, tutorial o immagini, per rendere più semplice l'apprendimento; in questo caso, poi, sarebbe stato ancora più gradito vista la difficoltà del software.
I manuali, ormai l'hanno capito quasi tutti, andrebbero sempre rilegati con una spirale, in modo da poterli tenere ben aperti semplicemente appoggiandoli; questi invece sono rilegati come quaderni e presentano una fastidiosa ostinazione a richiudersi mentre si è nel mezzo di un tutorial!
Il dischetto contiene un file compattato LZH, ma la procedura di decompattazione/installazione è completamente automatica e pertanto molto semplice.
In questa fase si può decidere di far copiare, nella propria directory (o dischetto) di lavoro, la versione normale, quella FP (Floating Point, per i fortunati possessori di 68020 o 030), o entrambe.
A installazione avvenuta, ci troveremo il programma vero e proprio e due directory, una per le nove texture, e una per i tre effetti speciali, di cui parleremo in seguito.
Anche se non si possiede un hard disk, non dovrerbbero sorgere problemi perché, come avverte cortesemente la documentazione, tutte

Una conchiglia modellata nel Detail Editor (immagine di Marco Pugliese)

Lo Script Editor di un'animazione di 17 Frame

E' facile creare forme bizzarre, un po' meno capire a cosa possano servire

Un primo piano del mostro!

le informazioni dovrebbero stare senza problemi anche su un dischetto.

La documentazione

Veniamo ora all'unico vero problema (le precedenti erano solo considerazioni) di questo pacchetto. Come i possessori di Turbo Silver ricorderanno, la Impulse non ha mai brillato per la documentazione fornita ai suoi (peraltro ottimi) programmi. Le potenzialità di questo sofisticato software (ben più potente, infatti, di Turbo Silver) non sono che minimamente accennate negli scarsi manuali in dotazione e possono essere rivelate, e sfruttate, solo con un paziente lavoro di "prova e riprova". Vengono forniti due manuali: una Reference Guide, di 78 pagine, che, proprio per il fatto di essere solo un manuale di riferimento molto sintetico, è da usarsi, pertanto, per "rinfrescarsi" la memoria, e una User Guide, sotto forma di Tutorial. Quest'ultima si presenta in 75 pagine, divise tra 18 tutorial, ma non risulta certamente sufficiente a far prendere la necessaria confidenza con la potenza del programma: effetti spettacolari come le texture o il Brush Wrapping, presente in ben 4 forme, vengono trattati in un paio di pagine. Anche i tutorial non sono poi molto curati. Un esempio: nel sesto ("Spin vs Sweep") si spiega che ci sono due modi per ottenere solidi di rotazione, spin e sweep appunto, e si aggiunge che è molto importante capire la differenza tra i due per poter ottenere buoni risultati. Segue una paginetta in cui si viene guidati nella creazione di due solidi con queste due funzioni e... stop! La differenza qual è?
Il libro non lo dice, bisogna capirlo da soli (praticamente lo Spin chiude sempre e comunque l'oggetto, anche se il punto più alto non si trova sull'asse di rotazione).
Sono d'accordo anch'io che basta farlo per capirlo, ma quantomeno denota o un'infelice idea del curatore del manuale o superficialità nella stesura della documentazione che, in un buon pacchetto commeciale, DEVE essere curata tanto quanto il programma, e, anzi, dovrebbe essere portata avanti parallelamente alla stesura del programma stesso. Questo software, per quanto costa e per quello che offre, ha il suo target d'utenza, almeno nelle intenzioni della Impulse, non certo nell'-hobbista quanto piuttosto in chi fa della produzione videografica, seppur a basso costo, un professionista insomma, che non può e non DEVE passare giornate intere a scoprire ciò che il manuale spiega male o addirittura non spiega neppure. Non è così che si aiuta Amiga a perdere l'etichetta di bel giocattolo ed entrare nel mercato del professionale, seppur, ripeto, nella fascia low-end, a bassissimo costo. A cosa serve scrivere un programma che "fa miracoli" se poi solo pochi riusciranno ad usarlo a pieno?
Del resto questo alla Impulse lo sanno, ma il "mea culpa" a pagina 1 del manuale, "vorremmo coprire ogni aspetto di questo programma, ma non ne abbiamo il tempo" (come non ne avete il tempo ?!?) non fa certo perdonare la ditta per il danno che arreca al suo stesso prodotto, e sono convinto che non sia un danno così marginale o minimo.
Quello di cui si sente la mancanza, per esempio, è una bella sezione dedicata ai parametri da assegnare alle superfici per simulare con realismo i materiali, quali vetro, cristalli, pietra o metalli. Sono tutti effetti che si possono raggiungere, ma solo provando e riprovando, e non è detto che tutti ce la facciano.
Tutti conosciamo le splendide immagini di Bradley Schenck, ma quanti di noi riescono a rifarle?

Il programma

Dopo questa "arringa", passiamo senz'altro alla decrizione del pro-

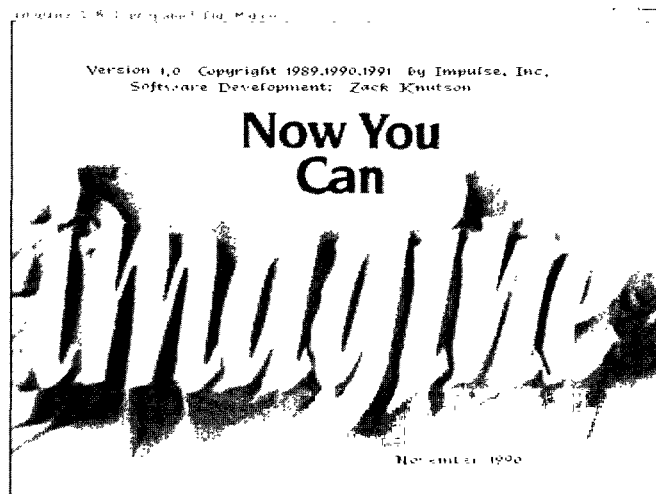
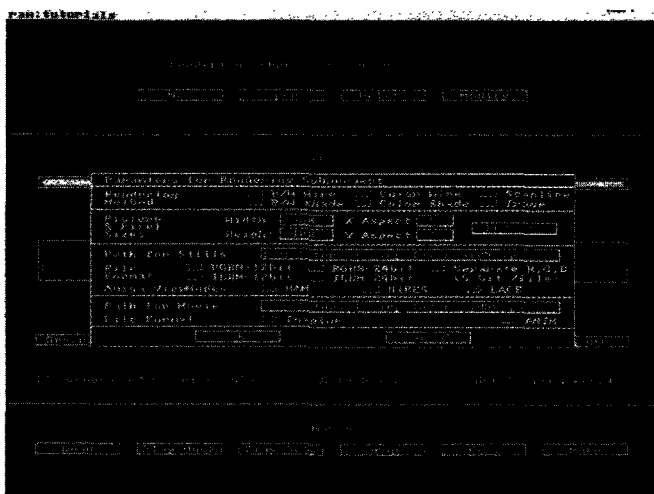
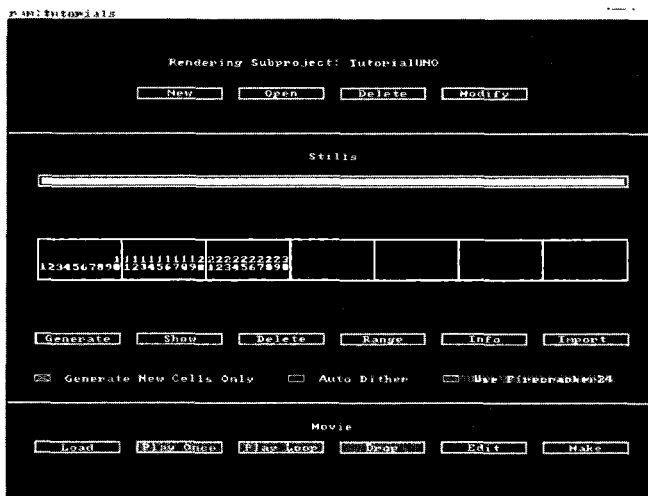
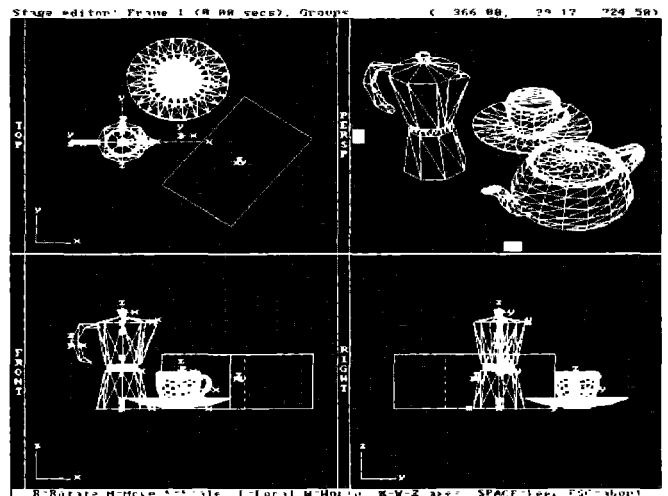
gramma.
L'ambiente di lavoro è stato suddiviso in cinque EDITOR, che vedremo in dettaglio, quattro dei quali relativi alla creazione degli oggetti e dell'animazione, molto intelligentemente caratterizzati dalla medesima interfaccia, cosa che rende il tutto molto coerente.
Le caratteristiche come i colori dello schermo, la risoluzione o le hot-key possono essere utilmente ridefinite attraverso la modifica del file di configurazione del programma (Imagine.config). L'ambiente di sviluppo, è la cosiddetta QUAD VIEW, dall'aspetto decisamente professionale: tre viste ortogonali e una in prospettiva, con punto di vista regolabile, il cui refresh avviene nei "tempi morti" dell'elaborazione e non intralcia, quindi, in nessun modo il lavoro.
Ognuna delle quattro viste può essere espansa a tutto schermo, per lavorare con maggior precisione, o ridimensionata alla grandezza di default, un quarto di schermo, con un semplice click.
Le tre viste ortogonali presentano, nei rispettivi angoli inferiori sinistri, gli assi del piano cui fanno riferimento, e confondersi diventa, quindi, più difficile, e possono essere dotate di griglia a passo regolabile; è sempre presente l'opzione di snap to grid per posizionare con precisione punti, assi o oggetti ed, per chi necessita di una precisione ancora maggiore, è sempre attivabile, nella barra di menu, un display con le coordinate attuali del puntatore.
Per quanto riguarda la finestra PERSPECTIVE, è possibile richiedere la vista in wireframe semplice, con rimozione delle linee nascoste o in shaded a 16 colori (solo a tutto schermo, però).
Anche con Imagine, come prima in Turbo Silver, è possibile lavorare su vari livelli (di astrazione oserei dire), e cioè su punti, linee, facce, oggetti o gruppi di oggetti.
Ora però ci sono nuovi modi per selezionarli: oltre a cliccare semplicemente sull'entita da selezionare, si può ora, con un Drag Box, aprire uno "scatolotto": tutto quello che vi è racchiuso rimarrà selezionato; in maniera analoga si può anche (udite udite!) disegnare a mano libera una linea chiusa che circonda ciò che interessa manipolare.Passiamo ora ad analizzare i quattro editor disponibili.

Project editor

Questa è la parte di programma da cui si creano, caricano o modificano i progetti, settandone tutti i parametri, quali risoluzione (fino a 8000x8000!), numero di colori (fino a 16.7 milioni, ovviamente non visualizzabili direttamente sullo schermo di Amiga) e formati di salvataggio (quelli Impulse o i normali IFF per le immagini o ANIM per le animazioni). E' anche supportata, per chi ce l'ha, la scheda a 24 bit plane della stessa Impulse, la Firecracker 24.

Forms editor

Questo è un editor di oggetti un po' particolare, e viene presentato, nel manuale, come una assoluta novità: permette di modellare le forme più strane, il tutto partendo sempre da una sfera. Sinceramente ne trovo l'utilizzo piuttosto difficoltoso e non sono riuscito a creare in effetti nulla di particolare, forse per la mancanza di idee; penso possa essere molto utile a chi ha idee stravaganti da realizzare: ripeto di non essere riuscito ad andare molto più in là di qualche strano vaso!
Comunque, nel manuale assicurano che può essere utilizzato per creare forme "organiche" complesse, esseri umani o asteroidi. Modellare forme "assurde", a caso, d'impeto, è estremamente facile, un po'

Il logo del programma

Il requester per settare risoluzione, colori ecc. delle immagini da realizzare

Il Project Editor, con pronta un'animazione di 30 frame

Lo Stage Editor con una scena olometica

meno, forse, realizzare qualcosa di cui si ha un'idea ben precisa.

Detail editor

Questa è senz'altro la parte di programma che si deve padroneggiare meglio, perché è quella in cui si passerà gran parte del tempo: è quella, infatti, in cui si creano oggetti (più tradizionali), o si modificano quelli creati nel Forms Editor. Possono essere caricati in questo modulo anche i file creati con Turbo Silver (versione 3 e seguenti). E' senza dubbio fatto benissimo, completo di quella che è la dotazione "classica" di un modellatore poligonale, e ha alcune feature care agli utilizzatori di Turbo Silver, come l'estrusione lungo un path, più molte altre nuove e, alcune, davvero stupefacenti, che verrano dettagliate in seguito.
Oltre alla creazione dei modelli geometrici, in questo modulo vengono anche specificate tutte le caratteristiche di superficie degli oggetti, come il colore, le texture e tutti quei parametri che regolano l'interazione tra i modelli e la luce.
Per le trasformazioni di scala, le traslazioni e le rotazioni sono disponibili due vie: una numerica, di precisione, tramite il requester Transformations, e una interattiva, in cui l'entità su cui lavorare viene racchiusa in un box, che può essere manipolata in tempo reale.

Cycle editor

In questo modulo, si possono creare, con estrema facilità, animazioni gerarchiche, in cui, cioè, il movimento di alcune parti è legato a quello di altre, proprio come il movimento di una mano è legato a quello di tutto il braccio.
Si può procedere alla creazione di uno "scheletro", ai cui segmenti legare uno qualunque degli oggetti creati negli altri moduli del pacchetto, e animare, con la tecnica dei key frame, lo scheletro stesso, sfruttandone la struttura gerarchica impostata durante la sua creazione. L'insieme dei key frame può essere salvato come cycle object e utilizzato nell'animazione, che viene composta nello Stage Editor.

Stage editor

Questo ultimo modulo, è quello in cui si preparano le scene di cui effettuare il rendering e le animazioni; si settano, cioè, la posizione della camera, le luci, i parametri del "mondo" di Imagine (luce ambientale, colore dello sfondo, del cielo, allo zenith e all'orizzonte...), e, soprattutto, i movimenti degli attori. In questo pacchetto un attore può essere una luce, la camera, oltre che, naturalmente, un oggetto vero e proprio. Per di più ogni attore, oltre che spostarsi, può trasformarsi durante l'animazione, e così si può animare la camera, cambiandone la focale, le luci, cambiandone colore o tipo, addirittura le texture sugli oggetti; e questi ultimi, poi, possono subire metamorfosi per trasformarsi in qualcos'altro. In questo contesto la finestra Perspective può essere settata per fornire il punto di vista della camera. A questo proposito c'è da aggiungere che il tracking di un oggetto, imporre cioè che la camera lo inquadri, risulta più macchinoso che in Turbo Silver.
Il cuore dello Stage Editor è l'Action Script, che risulta uno strumento nel contempo immediato e molto completo, permettendo di avere sott'occhio tutti gli "attori" contemporaneamente; per ognuno di questi sono presenti sei sbarrette, denominate timelines, che ne controllano ogni aspetto (posizione, allineamento, effetti speciali, dimensioni ...) in diretta relazione con l'evoluzione nel tempo dell'animazione.

La fantascienza

Passiamo ora ad esaminare le caratteristiche che fanno di Imagine un

prodotto unico e che lo distinguono dai suoi predecessori Nel Detail Editor sono presenti

Convert ILBM si può caricare un'immagine IFF e convertirla in un oggetto 3D. E' praticamente quello che fanno varie programmi, come Pixel 3D o DigiWorks, ma è la prima volta che viene incorporato in un software di modellazione e rendering.

Conform to Sphere/to Cylinder permette di "piegare" un oggetto, adattandolo ad una superficie rispettivamente sferica o cilindrica Può essere utile per realizzare flying logos come nel telegiornale della RAI

Slice ora si possono effettuare operazioni booleane tra solidi, funzione un tempo ad esclusivo appannaggio dei modellatori di superficie e mai prima d'ora apparso su un programma per Amiga (con l'unica eccezione per il, purtroppo, quasi sconosciuto Real 3D) Modellare una fetta di formaggio con i buchi diventa ora possibile, intersecando un parallelepipedo con una o più sfere, e buttando i pezzi di troppo Senz'altro è più facile a farsi che a dirsi

Brush Wrapping oltre al consueto color mapping che consente di avvolgere un'immagine IFF su un oggetto per modificarne il colore (esattamente come in Turbo Silver), Imagine offre altri tre tipi di brushing

- Altitude Map nella letteratura classica si è sempre chiamato Bump Mapping, ma è sempre bene cercare di confondere la gente, no? Questa opzione consente di applicare un'immagine IFF in scala di grigi, per modificare, in ogni punto, la geometria dell'oggetto Un esempio l' immagine di un volto, applicata ad una superficie, permette di avere un bassorilievo del volto stesso, con una "sporgenza" proporzionale alla tonalità di grigio usata
In effetti però non si ha una modifica del modello geometrico dell'oggetto; l'effetto viene calcolato in fase di rendering, con uno spostamento della superficie lungo la sua normale Questo si traduce nel fatto che, per esempio, nel realizzare un'arancia, applicando la giusta immagine della buccia su una sfera arancione, le asperità della buccia stessa non saranno visibili sul bordo dell'oggetto, che rimarrà perfettamente liscio perché liscia è la sfera usata

- Filter Map con questo effetto si possono simulare, per esempio, le vetrate dipinte delle chiese
- Reflect Map la funzione svolta non mi è per nulla chiara!

Il Brush Wrapping, putroppo è una di quelle feature mal documentate e il suo utilizzo risulta decisamente difficile

Dallo Stage Editor (menu Action) sono disponibili

Environment Mapping questa opzione consente di utilizzare un'immagine IFF come sfondo, e (qui sta il bello!) questa verrà riflessa negli oggetti della scena che presentano

La Teiera di Martin Newell

Chiunque si sia occupato un po' seriamente di Computer Grafica, si sarà certamente imbattuto spesso in un oggetto un po' particolare: una teiera, the teapot, anche nota come Utah teapot o anche Newell' s teapot. Se ne sono viste di tutti i colori, con tutte le texture immaginabili, alcune addirittura "gommose", come quelle che rimbalzavano e si rompevano in un recente video IBM, ma un cosa è rimasta (quasi) sempre la stessa· la forma Qualcuno ne sarà ossessionato, qualcun'altro si è affezionato a questo oggetto così semplice.

Un po' di storia

Nel 1971, nella Utah University a Salt Lake City, sotto la direzione di Ivan Sutherland, uno dei padri, forse IL padre della Computer Grafica (e della realtà virtuale), ferveva la ricerca: si studiavano e sviluppavano algoritmi di rendering, nulla però di neppur lontanamente paragonabile al quel Ray Tracing cui siamo ormai abituati. Stanchi di usare cubi, tubi o sfere (queste ultime ancora purtroppo decisamente amate da molti presunti "professionisti dell'immagine" che lavorano su Amiga), molti sforzi vennero spesi nel creare modelli interessanti su cui testare i loro nuovi algoritmi Sutherland stesso girò per settimane per Salt Lake City con la sua Volkswagen Maggiolino dipinta con i punti di riferimento che i suoi studenti, durante le ore di lezione, utilizzavano per ricostruire il modello geometrico dell'auto. Con questo stesso spirito, nel 1975, Martin Newell studente presso quella stessa università, notò l'attraente forma della teiera poggiata sulla sua scrivania. Dopo averne fatto, a mano, un bozzetto su carta millimetrata, Newell trovò, a occhio, i punti di controllo per le superfici di Bezier che avrebbe usato per modellarla. Nacque così la Utah teapot, che Newell usò nella sua tesi di laurea e che consisteva originariamente di 28 patch, ora portati a 32 con l'aggiunta del fondo (altrimenti il the cadrebbe!). La vera teiera, insieme agli schizzi di Newell e alle immagini della sua tesi, sono conservate al Computer Museum di Boston, Massachusset; chi avrà la fortuna di vederla, noterà che la sua forma è un po' diversa da quella a cui siamo stati abituati: la versione attuale è infatti alta solo l'80 % dell'originale, modifica introdotta da Jim Blinn, altro "guru" della Computer Grafica, per renderla ancora più piacevole.
La teiera è stata, da allora, utilizzata come benchmark per testare nuovi algoritmi di rendering, poiché la sua superficie presenta curvature positive e negative, e intersezioni da gestire correttamente. Il principale "colpevole" della diffusione di questo oggetto è comunque il già citato Blinn, piuttosto che Newell stesso, che ne ha realizzate innumerevoli immagini Chi la odia potrà comunque facilmente farla esplodere con Imagine!

caratteristiche di riflessione E' un effetto molto interessante perché funziona anche anche in modo Scanline che generalmente, essendo molto più veloce del Full Trace, non può manipolare riflessioni Anche questo, putroppo, è decisamente complicato da usare e i risultati dipendono solo dalle ore spese in tentativi.

F/X. Questa è una delle potenzialità più interessanti offerte da Imagine con questa opzione si caricano delle routine esterne, dei veri e propri programmi, che risiedono nella directory FX, e che creano, appunto, degli effetti speciali per le animazioni Quelli forniti di serie sono tre·

-Explode. permette di far esplodere un oggetto, tramite un progressivo allontanamento delle singole superfici che lo compongono Ci sono dei parametri da settare per determinare il tipo di esplosione

-Ripple: consente di creare increspature come quelle dell'acqua quando vi cade un oggetto dentro

-Grow permette di far letteralmente "crescere" un oggetto, lungo un percorso definito Sarebbe bello che la Impulse mettesse a disposizione la documentazione necessaria per poter scrivere questi effetti

Impressione d'uso

Come sarà trasparso dai paragrafi precedenti, quello che si prova nell'utilizzare questo software è essenzialmente (complice anche il non aver potuto spremerlo per molto tempo) frustrazione. Frustrazione per il fatto di sentire di avere sotto le mani (o meglio, sotto il mouse!) una potenza davvero grande, ma di non avere la possibilità di poterla liberare se non grazie ad un miscuglio di fortuna, caso e intuizione, tutte cose che poco si addicono ad un contesto rigidamente deterministico, come quello informatico Il programma molte volte sembra "non gurdare" dei parametri che gli vengono forniti, sembra quasi dotato di una sua personalità tutt'altro che conciliante, quanto piuttosto testarda, e senz'altro richiede molto molto tempo Probabilmente molto si può ancora fare per renderlo più friendly, se non altro una capillare opera di divulgazione (di trucchi, diciamocelo!) a riguardo E' fuori dubbio la qualità, ma penso che questo pacchetto meriti ancora del lavoro da parte del team che l'ha realizzato. Di sicuro c'è chi riuscirà a "estorcere" immagini o animazioni molto belle da Imagine; è un peccato che quasi sicuramente vorrà tenere (stupidamente) per sé questi suoi segreti

Una cosa invece totalmente assente e che, penso, non può più mancare in un prodotto che vuole essere professionale è un'interfaccia ARexx Senz'altro chi ha bisogno di un rendering di qualità e non è spaventato per il lavoro che lo attende, magari perché si è già "fatto le ossa" su Turbo Silver, può comprare Imagine senza temere nulla· ne vale la pena ! Almeno fino all'uscita di RenderMan, se mai avverrà ▲

Imagine è in vendita presso:
Alex Computer
C.so Francia, 333/4 - Torino
Tel. 011-7730184-4033529

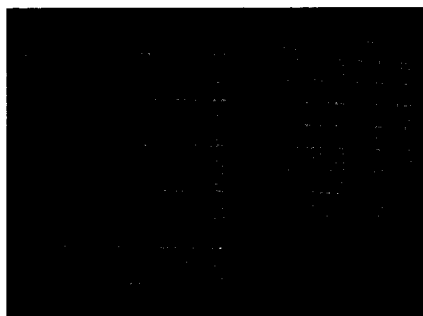
Totomania

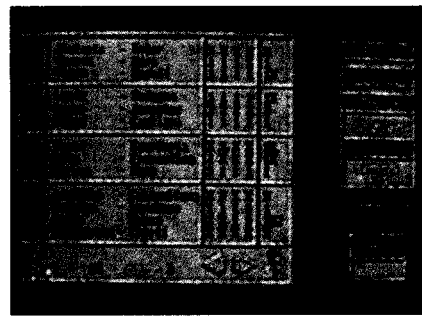
Marco Tortolina & Mirco Baiardi

Già da qualche anno, è grande il successo tributato agli elaboratori dagli appassionati giocatori di schedine Ormai anche il sistemista più sprovveduto si aiuta con il computer, al fine di ottenere le maggiori probabilità di vincita con la minor spesa possibile Il parco software dedicato è veramente nutrito si parte dai vecchi e semplici riduttori normali (molti si trovano di pubblico dominio) e si finisce con i pacchetti super professionali (solitamente usati nelle sole ricevitorie) capaci di imporre ai sistemi notevoli e sofisticate riduzioni Per molti versi Amiga è rimasto, sino ad oggi, un po' ai margini di questa attività, forse per le sue caratteristiche prettamente orientate verso la grafica e anche, diciamolo pure, perché i programmatori italiani impegnati nel settore hanno finora preferito lavorare con i tradizionali e più conosciuti PC Solo negli ultimi tempi si è potuto notare in questo campo un certo fermento e anche sui periodici specializzati nella sistemistica si incominciano a vedere le pubblicità dei programmi Totocalcio per Amiga Tra questi ricordiamo TOTOAMIGA e SISTHEMA, disponibili in versioni amatoriali e professionali, queste ultime si differenziano dalle prime solitamente dalla possibilità di accedere alla stampa diretta sulle schedine.

Totomania

Il pacchetto che prendiamo qui in esame è TOTOMANIA della software house italiana IDEA La confezione, con raffigurato un benaugurante gruzzolo di soldi, contiene il disco programma e dati e un buon manuale completato da alcune nozioni sugli sviluppi dei sistemi. Il capitolo di introduzione ci dice che si dovrebbero scollegare dal computer gli eventuali drive e le periferiche, la cosa ci lascia un po' perplessi ma infiliamo lo stesso il disco nel drive senza aver modificato nulla dal nostro Amiga comprendente hard disk, memoria, stampante e secondo drive Funziona tutto e dallo schermo Workbench appaiono le icone dei due tool principali TOTOMANIA, per elaborare, e STAMPA, per stampare su tabulato i dati ottenuti Un doppio click fa "scattare" TOTOMANIA che ci porta ad un requester che richiede una parola di una certa linea del manuale è la chiave per entrare, provate a sbagliarla e il vostro monitor comincerà ad "urlare". Si accede così al primo dei quattro schermi, collegati tra loro, di cui è dotato TOTOMANIA E' in pratica l'editor del sistema da realizzare e la prima cosa che ci balza all'attenzione, è che non si tratta della normale mascherina che accetta i segni 1X2 Dopo, infatti, avere inserito i nomi delle squadre che compongono la schedina, l'editor attende l'inserimento delle sole percentuali di successo dei tre segni (detto picchetto e riportato dai vari giornali specializzati) e in base a queste formerà automaticamente un primo sistema integrale. Le percentuali possono essere calcolate anche automaticamente, inserendo i tabellini delle squadre partecipanti, per intenderci i tabellini sono i dati riportati da tutti i quotidiani sportivi sulle partite disputate vinte e perse in casa e fuori dalle varie squadre. Il tutto si muove tramite una buona interfaccia utente degna di Amiga che solo in alcuni casi pare un po' troppo appesantita da eccessivi preziosismi e limitata dalla programmazione realizzata in BASIC compilato. Terminata la prima fase, ci si può apprestare a snellire il sistema tramite una prima riduzione. L'elaborazione è determinata dalle sezioni, il cui numero non può essere superiore a tre (dividono in prati-

ca le colonne) e dal numero delle triple a queste assegnate (sino a tredici). TOTOMANIA si basa per l'input solo sulle triple e non sulle fisse e doppie che vengono però determinate in base ai valori percentuali dalla particolare elaborazione del sistema. L'opzione ELABORA ci dà modo di visualizzare le schedine e le colonne, con i relativi costi, portandoci nella seconda schermata. Da questa si potrà accedere all'archiviazione su disco oppure, attivando con un gadget il terzo pannello, ad ulteriori riduzioni del sistema mediante il condizionamento del numero dei segni, dei consecutivi e delle interruzioni. Da notare che prima di decidere di archiviare il progetto del sistema, bisogna accertarsi di non essere in condizione di dover apportare modifiche, in quanto non sarà più possibile editarlo ma solamente stamparne le caratteristiche con il tool di stampa o passarlo allo spoglio elettronico. Lo spoglio è il compito del quarto schermo di TOTOMANIA, indispensabile specialmente se si sono giocate molte colonne, ci fornisce, dopo l'introduzione della colonna vincente, tutti i punteggi ottenuti e su quale schedina e colonna sono stati ottenuti.

Conclusioni

Migliorabile senza dubbio in alcune sue parti, come, ad esempio, l'accesso all'archiviazione e la gestione dei device (non è infatti prevista alcuna scelta di percorso tramite requester), TOTOMANIA si presenta ad un pubblico senza eccessive

pretese come un buon programma molto semplice da usare anche per chi di Totocalcio ne capisce poco. E' necessario constatare, inoltre, che il BASIC compilato con cui è stato programmato TOTOMANIA non spicca certo per le sue capacità di sfruttare al meglio le qualità della macchina. E' quindi auspicabile che le caratteristiche interessanti del programma vengano un giorno migliorate e ampliate con linguaggi decisamente più idonei a moderne tecniche di programmazione. ▲

Totomania è distribuito da:
Leader Distribuzione

(segue da pag. 60)

ampliando scrivendoci). L'ultima funzione, che illustreremo in loco, è quella di ricerca, ci permette, sostanzialmente, di posizionarci in prossimità di un record particolare. Per smaltire tutte le calorie ingerite con i nostri esempi gastronomici, ci siamo iscritti ad una corsa podistica, è arrivato primo il concorrente numero 1753, la giuria guarda nello schedario dei concorrenti e trova la cartella a lui associata. Vi sono due modi, fondamentalmente, per svolgere un compito di questo genere, cercare su tutte le schede dalla prima all'ultima il numero 1753, o selezionarla direttamente. Noi umani, purtroppo, non abbiamo una precisione tale da riuscire, con un solo colpo d'occhio, a capire dove si trova la scheda con precisione, al contrario i calcolatori sono molto potenti in questo tipo di operazione. Qual è, allora, il comando che dice al nostro Amiga di posizionarsi in corrispondenza del record in questione? Se avete pensato a find avete sbagliato, il comando è Seek:

```
posizione=Seek
(file_handle,nuova_posizione,modo)
```

Notiamo che esiste una stretta cooperazione tra nuova_posizione e il modo di accesso, poiché nuova_posizione è uno spiazzamento rispetto ad una delle tre posizioni fondamentali e che sono i valori che può avere modo:

```
OFFSET_BEGINNING
```

quindi lo spiazzamento dall'inizio del file:

```
OFFSET_CURRENT
```

quindi lo spiazzamento dalla posizione corrente:

```
OFFSET_END
```

lo spiazzamento dalla fine del file. Facciamo un breve esempio di scrittura su un file: il nostro programma apre una finestra console e scrive una frase.

```
#include "exec/types.h"
#include "exec/nodes.h"
#include "exec/lists.h"
#include "exec/libraries.h"
#include "exec/ports.h"
#include "exec/interrupts.h"
#include "exec/io.h"
#include "exec/memory.h"
#include "libraries/dos.h"
#include "libraries/dosextens.h"

#define AMPIEZZA_BUFFER 256
main()
{
 struct FileHandle *esempio;
  if((esempio=Open("CON:0,0,200,100/
ciao",MODE_OLDFILE))==0
    Exit();
   Write (esempio,"Amiga Magazi-
ne",14);

   Close(esempio);
}
```

▲

Disk Drive Master 3A-1D

Marco Zandonadi

La Golden Image è una ditta produttrice di hardware che si sta facendo notare da qualche tempo, grazie ad una campagna pubblicitaria piuttosto consistente
La gamma di prodotti di questa casa è abbastanza varia e riguarda più di una macchina (Amiga, Atari ST, IBM e compatibili, ecc.) Ecco alcuni tra gli articoli più interessanti per Amiga il mouse ottico GI-1000 (che recensiremo sul prossimo numero di Amiga Magazine), lo hand scanner JS-105-1M (le cui prestazioni arrivano fino a 400 dpi e 64 toni di grigio), le espansioni di memoria RC-500 e RC-2000 (rispettivamente per il 500 e il 2000) e, infine, i disk drive esterni Master 5A-1 (da 5 pollici e 1/4), 3A-1 (da 3 pollici e 1/2) e 3A-1D (identico al precedente ma con l'aggiunta del track display)
In questo articolo ci occuperemo proprio del Master 3A-1D.

La confezione

Nella confezione (che è ben protetta da uno spesso strato di polistirolo) trovano posto il drive, la cartolina di garanzia, un manuale in tre lingue (tra cui, naturalmente, non c'è l'italiano..)

Track display e volume

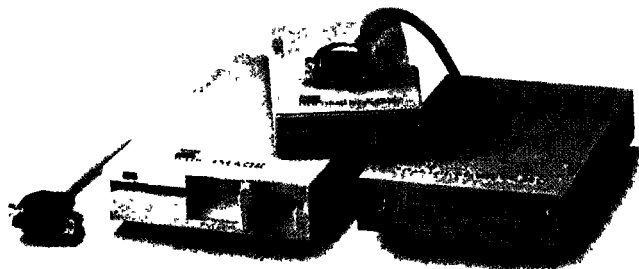
La prima impressione che si ha prendendo in mano il Master, è piuttosto strana: voluminosità e leggerezza al tempo stesso.
Se paragonato al drive esterno slim line, il 3A-1D occupa circa il doppio dello spazio pesando la metà! La responsabilità di tanto volume in più, va imputata, senza dubbio, alla presenza del track display un piccolo schermo digitale che ha il solo scopo di indicare la traccia sulla quale si trova la testina Non sono affatto convinto che la presenza di questo gadget sia più importante di fattori ergonomici come le dimensioni del drive Quanto detto è avvalorato dal fatto che il 3A-1 (cioè la versione del 3A-1D senza track display) ha le dimensioni di un comune slim line

I pregi: silenziosità ed estetica

A parte l'eccessiva voluminosità, il Master è un drive di buona fattura e non manca di evidenziare le sue qualità durante l'uso Innanzitutto è silenzioso Questa, vi assicuro, è la migliore dote del 3A-1D Il rumore emesso è ridotto ad un lieve sussurro meccanico una vera favola se paragonato al frullatore che c'è al posto dell'unità dischi interna degli Amiga 500! Un altro punto a favore del 3A-1D è la sua linea estetica: semplice ed elegante (anche se non è molto vicina a quella di Amiga). Infine, il Master, al contrario dei drive originali Commodore, dà un'idea di robustezza e compattezza: niente tasti per l'espulsione che si rompono (come nei 500) né parti di plastica morbida (come nei 2000) Non ha dato particolari problemi nel corso della settimana in cui abbiamo potuto testarlo

Annotazioni finali

E veniamo al prezzo Il 3A-1D costa attorno alle 195 000 lire mentre il 3A-1 circa 160 000 La differenza di prezzo (come quella di volume) è dovuta allo stravagante Track Display (forse qualcuno comincerà a sospettare che non mi sia particolarmente simpatico), che, ripeto, non ha altri scopi che quello di indicare il numero della traccia su cui sta passando la testina
A parte queste stranezze, il Master si è rivelato un buon prodotto: se state pensando di acquistare un drive esterno, il 3A-1D potrebbe essere la scelta giusta ▲

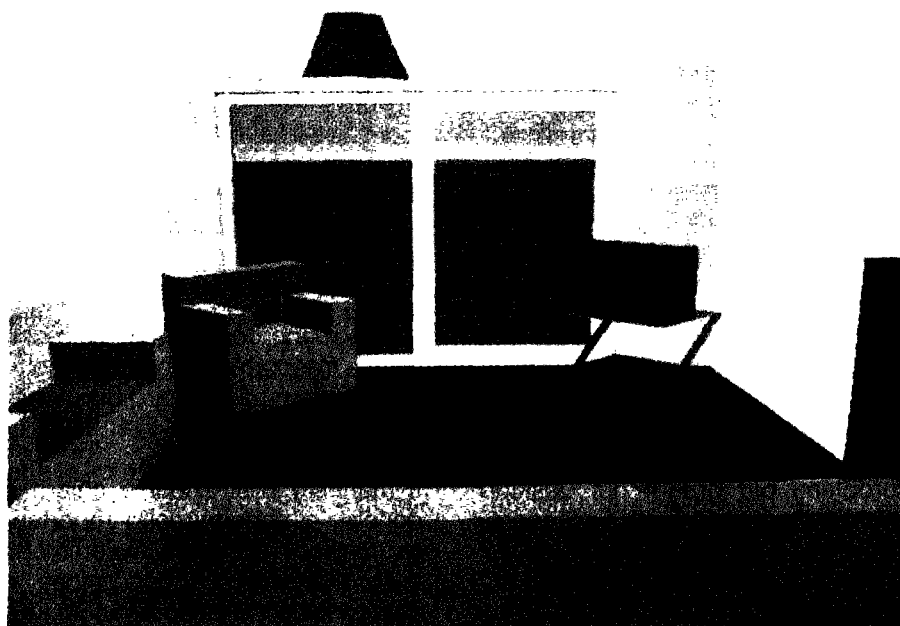
Il Master 3A-1D è distribuito da:
Armonia Computers s.r.l.
Via Conegliano, 33 - Susegana (TV)
Tel. 0438/435010

La serie dei drive della Golden Image per Amiga, tra cui il Master 3A-1D

Arriva la Realtà Virtuale

Marshal M. Rosenthal
Photo: Marshal M. Rosenthal NYC

Povero Marshal Costretto ad abbandonare l'eccitante Las Vegas Convention Center, senza tempo per mangiare, afferrando un taxi in mezzo ad una folla di assalitori con la medesima intenzione, passando davanti a ristorante dopo ristorante (ma non era l'Hard Rock Cafe?) e tutto perché la Psygnosis ha preferito esporre i propri prodotti in un hotel invece che in fiera
Marshal finisce all'Alexis Hotel, non uno di quelli che è facile attraversare, con *ascensori e cose simili*, NOOOO, è uno di quelli piuttosto eleganti costruiti attorno ad un campo da golf
Lo stomaco di Marshal ruggisce, Marshal ha la nausea, Marshal trova finalmente la porta giusta è chiusa Marshal vuole andare a casa o almeno trovare un bar
Ebbene, che ha a che fare tutto questo con la realtà virtuale, vi chiederete? Ci arriviamo quaranta minuti più tardi il nostro uomo si trova a divorare con gli occhi la nuova linea di prodotti Psygnosis, trangugia la sua sesta tazzina giornaliera di high-test (caffè nero, saturo di caffeina), sono solo le 11 antimeridiane, e infine posa le sue stanche membra sullo sgabello di un bar, vicino al vassoio con le ultime brioche.
E' adesso che le cose diventano interessanti Non essendo i suoi sensi del tutto intorpiditi (ed essendo lo stomaco occupato con qualcosa), Marshal riesce a notare un tipo in un angolo, lontano dai computer, che tiene in mano un Watchman Sony, uno di quei piccoli VCR da 8 mm
La curiosità ha ucciso molti giornalisti, ma non c'è modo che il nostro uomo desista dall'idea di osservarlo più da vicino, specialmente perché c'è qualcosa di interessante nello sfarfallio di quello schermo LCD Le presentazioni sono accompagnate da uno sguardo che rimane fisso sullo schermo
Il risultato è che Ian Andrew, presidente della Incentive Software, comincia a raccontare molte cose su quello che stanno vedendo
Si tratta di Realtà Virtuale (l'avevo detto che eravamo arrivati al punto)
Il team di progettisti di Ian, è il responsabile della creazione del Castle Master, l'adventure in grafica solida 3-D che viene commercializzato *con il nome di Domark in Gran* Bretagna e nel resto d'Europa Castle Master è stato creato con un sistema grafico tridimensionale chiamato The 3-D Construction Kit, che ora viene venduto con questo nome
Il Kit impressiona favorevolmente, sebbene non sia ancora nulla rispetto a ciò che Ian sta vedendo sul quel Watchman
Ma prima il Kit
"Il sistema è stato progettato sulla base del Freescape 2 di nostra proprietà e permette la creazione e la costruzione di una realtà virtuale" afferma Ian
"E' il primo prodotto che consenta di creare, progettare e pianificare un ambiente reale e poi di camminare in quell'ambiente e interagire con esso come se si fosse veramente al *suo interno*"
"Usare il Kit è come costruire con dei blocchi, sebbene esista un'interfac-

cia grafica che rende le cose molto più semplici. Fate un muro, lo deformate, vi stringete ed entrate nel mondo che avete costruito; animate gli oggetti e li fate muovere entro di esso".
Ian sottolinea come le diverse "primitive" (sagome) consentano di creare edifici, strade, sotterranei, persino veicoli ordinari come le automobili o astratti come le sfere volanti.
E' presente anche una libreria di effetti sonori pronta per l'uso. Il programma realizza grafica solida tridimensionale con rotazioni e movimenti molto rapidi, sia con macchine a 8 che a 16 bit.
Conoscendo il pubblico, è stato aggiunto un gioco arcade/adventure scritto con il Kit.
Non solo dimostra gli effetti e le animazioni che possono essere realizzate, ma risulta anche divertente.
Ora, Marshal è disposto ad ammettere che tutto ciò appare molto bello, dopo tutto Castle Master fa un certo effetto, ma la Realtà Virtuale implica anche l'impostazione di parametri che operino indipendentemente dall'osservatore, o Dio, se lo si preferisce.
Cose come la gravità, o ciò che accade quando una nuvola copre il sole, le persone e così via. Tutti aspetti che non possono essere modellati in maniera adeguata su un sistema a 16 bit come Amiga.
Questo tipo di cose richiede una grande potenza.
Ma torniamo a ciò che aveva attratto Marshal al principio.
A un primo sguardo poteva sembrare The 3D Constrution Kit, ci sono delle somiglianze, ma c'erano anche degli aspetti che indicavano che si trattava di una faccenda del tutto diversa.
Ora non fatevi confondere dal fatto che Ian si travesta: ciò che stava guardando era Supercase Virtual Realities, sotto gli auspici della Dimension International (si tratta ancora di Ian e del suo team). Questo sistema è stato progettato per essere attivato secondo le necessità dell'utente; questo può essere un individuo, un'azienda, o qualsiasi altro soggetto.
Ciò che l'utente ottiene è una Realtà Virtuale personalizzata, un sistema completo, comprendente tutto il software e l'hardware necessario, assieme ad una implementazione a tre dimensioni dell'ambiente richiesto, sia che già esista o meno.
L'utente può poi muoversi al suo interno, interagire con questo mondo virtuale e vedere le cose come se si trovasse realmente in esso, aiutato in questo da un dispositivo di controllo a sei direzioni. Il tutto con un'accuratezza estrema e in tempo reale.
Mentre osserva questo mondo artificale (dall'interno di una vistosa auto sportiva rossa, ovviamente), Ian mi confida alcune delle condizioni che l'hanno reso possibile.
"Il tipo di potenza grafica necessaria alla Realtà Virtuale non esisteva 10 anni fa" mi dice, "almeno non per coloro che non avevano accesso ad un super-computer, di quelli gelosamente conservati dal governo. Un sistema personalizzato fondato su un PC IBM 486 è ora in grado di eseguire i milioni di calcoli necessari, quando sia accompagnato dal nostro software e da una scheda grafica di alta qualità".
Risoluzione dell'ambiente? Sorride: "Immagina la grandezza della Gran Bretagna con l'accuratezza di un millimetro.
Si possono collegare allo stesso ambiente fino a 8 terminali e ciascun operatore può avere un controllo indipendente degli oggetti selezionati e/o dei punti di vista".
Ian sottolinea che molte delle carat-

teristiche devono essere viste per poter essere apprezzate.
"Cose come il movimento di più oggetti sotto il diretto controllo dell'utente, o mediante modelli predeterminati, o mediante controllo intelligente.
Ci sono opzioni per punti di vista multipli, sia statici che in movimento e con la possibilità di associarli a oggetti in movimento, zoom e blocco del punto di vista".
L'elenco continua con animazioni umani e animali, velocità lineari e angolari, specifiche di trasparenza, scelta della gravità e dell'attrito.
Ian prosegue: "Si tenga presente che tutto ciò è completamente interattivo, ci si può muovere a piacimento, aprire e chiudere porte, accendere le luci, entrare in veicoli e palazzi.
Il che significa che è possibile guidare un'auto, che si può camminare, correre o volare cambiando il punto di vista come meglio si preferisce".
Ferma il registratore e si sposta, con l'avvolgimento veloce, in un'altra sezione.
Questa visualizza un certo numero di figure umane che corrono, muovendosi attraverso un paesaggio costituito da un mosaico colorato casualmente, quasi come se una trapunta gigante fosse stata buttata sulla Terra.
I corridori sono bellissimi, con movimenti naturali delle articolazioni ed espansione del torace durante il movimento.
Ci si è sforzati di dare il senso della realtà mediante i dettagli più infimi, sia che si tratti di uomini che di oggetti.
Naturalmente, è possibile fare cose impossibili nel mondo cosiddetto "reale", come alterare il proprio punto di vista.
In questo caso noi ci spostiamo da un corridore all'altro.
Ma potremmo anche prendere un'auto e osservala avvicinarsi a un edificio particolare, poi spostarci al suo interno e osservare l'auto dall'esterno, attraverso la finestra che le sta di fronte.
Girarci per trovarci in un ufficio, la

cui stanza ha il solito numero di sedie standard e di tediose scrivanie, con qualche pianta e un quadro sul muro antistante (un particolare notevole: lo schermo del PC sulla scrivania si illumina quando viene acceso).
Ian fa avanzare ancora il nastro fino ad un punto dove, per divertimento, l'utente ha azzerato la gravità. Ooops, tutto comincia a fluttuare, anche gli edifici.
Gli usi per la Realta Virtuale abbondano.
Provate a pensare a un museo interattivo, che mostri oggetti, scene, strutture e costruzioni altrimenti impraticabili o impossibili da rappresentare.
Oppure, come familiarizzarsi meglio con una regione o un continente lontano, mesi o anni prima di recarvisi?
O, ancora, al lavoro su una piattaforma spaziale.
O camminare entro l'ala che deve essere ancora terminata della vostra nuova casa e vedere come interagisce con il resto dell'ambiente.
I pianificatori urbanistici esulterebbero se avessero a disposizione un tale strumento per verificare i cicli vitali delle aree metropolitane.
Per non dimenticare gli uomini della pubblicità e delle pubbliche relazioni, che scherzerebbero per giorni pensando al potenziale di vendita.
Ma persino l'high-tech giapponese ha bisogno di batterie nuove e pertanto lo schermo si oscura. Come pure la capacità di Marshal di ritrovare la strada attraverso il labirinto di suite per tornare al CES. ▲

Biografia dell'autore:

Marshal M. Rosenthal è uno scrittore e un fotografo-giornalista, residente a New York, il cui lavoro lo porta in tutto il mondo. Non ha ancora completamente realizzato il suo sogno, quello di diventare anche una rock star.

Prodotti:

Incentive Software
Domark LTD.
3D Costruction Kit

Dimension International
Zephyr One, Calleva Park
Aldermaston
Berkshire, England RG7 4QW

Hand Scanner JS-105-1M

Derek Dela Fuente
British Correspondent

Il JS-105-1M è uno scanner manuale prodotto dalla Golden Image che permette di catturare immagini e testo senza grosse difficoltà, sormontando così uno dei problemi più grandi del desktop publishing e della grafica in generale.

Il pacchetto

Lo scanner viene fornito in una elegante scatola contenente il manuale di installazione hardware, un manuale per il software, un'interfaccia, un alimentatore e un disco 3"1/2 con il software di gestione dello scanner.
L'installazione richiede un attimo di attenzione: dopo aver collegato il cavo proveniente dall'interfaccia alla porta parallela, è necessario collegare il cavo dello scanner all'interfaccia alla quale deve essere connesso il cavo proveniente dall'alimentatore.
Quest'ultimo, inoltre, deve essere collegato alla presa di rete per ultimo. L'installazione software su hard disk è meno complicata, visto che basta trasportare l'icona-cassetto del programma sull'hard disk, operazione che, complessivamente, richiede qualche minuto.
Il manuale d'istruzioni è chiaro e conciso e, aspetto molto importante, fornisce parecchi avvertimenti sulle modalità di connessione e disconnessione dello scanner durante il funzionamento; il manuale ricorda, inoltre, che è possibile collegare anche dispositivi di switch alla porta parallela (per utilizzare, ad esempio, la stampante) e quale accorgimenti adottare in questi casi.

Funzionamento

Dopo aver impostato lo switch presente sul lato dello scanner (modalità di scansione: testo, 200, 300 o 400 dpi) e aver regolato il contrasto con l'apposita rotella, sempre presente sul lato dello scanner, si è pronti per catturare ciò che si desidera: con il mouse bisogna selezionare nel programma l'icona raffigurante lo scanner e, dopo aver selezionato la larghezza e l'altezza dell'immagine da catturare, bisogna tener premuto il tasto presente sul lato sinistro dello scanner. A questo punto, non appena si accende la luce verde, si può iniziare a far scorrere lo scanner sull'immagine. Quando si spegne la luce verde, si può rilasciare il bottone e, dopo una breve attesa, ammirare l'immagine catturata sullo schermo.
Sullo scanner è presente, inoltre, un ulteriore led che ha un compito molto importante: avvertire l'utente, attraverso un lampeggio, che la scansione dell'immagine è troppo veloce e che è necessario rallentare, o (led spento) che sono stati persi dei dati durante questa operazione.
La larghezza dell'immagine è di 4 pollici, anche se con l'opzione multiscan è possibile arrivare a 12 pollici.
Bisogna segnalare però una nota dolente: lo scanner non è utilizzabile su A1000 nonostante negli Stati Uniti sia in commercio uno speciale adattatore che ovviamente non è disponibile in Italia.

Il software

Il pacchetto grafico di gestione dello scanner fornito nella confezione si chiama Touch Up ed è sicuramente uno tra i più potenti in commercio. Sebbene l'impatto grafico non fornisca questa impressione (l'abito non fa il monaco) il programma ha delle caratteristiche che non si trovano raggruppate in nessun altro software del genere.
L'unico neo del programma è l'este-

nuante lentezza di esecuzione delle operazioni anche più semplici, lentezza che si nota visibilmente soprattutto nell'operazione di refresh dello schermo.
Tutti i comandi standard caratteristici di un programma di grafica, sono presenti in forma molto semplice come la rotazione, lo stiramento e la distorsione di oggetti.
Il software contiene, inoltre, una sezione interamente dedicata alla grafica strutturata con funzioni come B/Spline, curve di Bezier, il riempimento di aree e altre caratteristiche, come la possibilità di selezionare una particolare area da utilizzare poi come brush e una funzione che permette di modificare facilmente bit per bit l'immagine. Un particolare che si nota facilmente è la differenza tra le icone del programma e quelle standard che vengono solitamente utilizzati in questi tipi di programmi.
Una delle caratteristiche più esclusive di Touch Up, è la possibilità di caricare e salvare le immagini in molti formati. I file monocromatici vengono caricati direttamente nel programma, mentre gli hi-res, medres, HAM, EHB e PC Paintbrush a colori vengono automaticamente convertiti in bianco e nero. Una volta che si ha finito di catturare le immagini, esse possono essere salvate in IFF, IMG, TIFF, MacPaint, PCX e GIF.
Un difetto del programma è che non ha alcuna possibilità di stampare le immagini, ossia si dà per scontato che le immagini vengono stampate con un programma di grafica diverso, come DPaint.

Configurazione

Hand Scanner della Golden Image richiede minimo un Amiga 500 con 1 MByte di RAM per funzionare correttamente. Se si prevede di dover catturare immagini molto grosse o intere pagine di testo, allora si deve disporre di 2 o, meglio, di 3 MByte.

Conclusioni

Lo scanner della Golden Image è sicuramente un dispositivo che si confronta bene con i concorrenti; si possono ottenere immediatamente ottimi risultati da semplici disegni, mentre è necessaria un po' di esperienza per immagini più complesse con sfumature.
Il prodotto è uno tra i più costosi della sua categoria, anche se il rapporto qualità prezzo è sicuramente superiore ai concorrenti. ▲

Il JS-105-1M è distribuito da:
Armonia Computers s.r.l.
Via Conegliano, 33 - Susegana (TV)- Tel.0438/435010

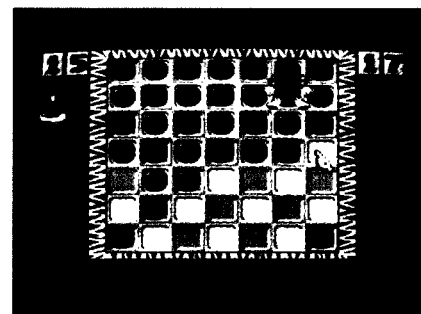
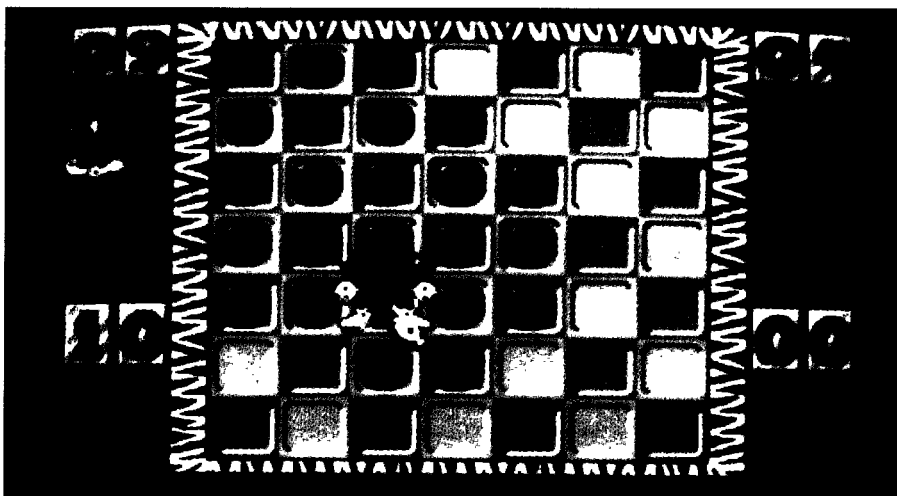
(segue da pag. 61)

tilmente ed efficientemente il reparto tecnico mi ha rispedito (a sue spese) il modem riparato in garanzia. Considerato che spedirlo lì mi è costato L.125.000 di corriere, non è poi così male. Da un punto di vista *puramente funzionale*, il DL2000 nella versione più avanzata (con MNP-5 e SendFax) è veramente un buon prodotto. Avevo alcune riserve, che però si sono sciolte con la release 1.3 del software di gestione. In particolare, l'MNP-5, un protocollo di compressione e controllo degli errori molto diffuso, permette di ottenere comunicazioni decenti anche sulle penose linee italiane. La possibilità di mandare FAX non l'ho mai sfruttata, ma potrebbe rivelarsi interessante. Purtroppo non sono riuscito ancora a capire la causa dei suoi strani comportamenti. La mia personale opinione, è che si tratti sempre di qualche efferatezza perpetrata dalla mia motherboard 6.1. I problemi si riscontrano infatti solo in fase di AutoConfig, vale a dire proprio quando la motherboard dà il peggio di se stessa (come ho già raccontato altrove, la serie 6.x con x<2 delle motherboard di Amiga 2000 ha grossi problemi di rumore spurio sul bus durante l'AutoConfig). Nessun altro utente di quelli che conosco ha riscontrato sintomi analoghi. Tra le varie caratteristiche del DL2000, spicca la possibilità di fissare la comunicazione modem <-> computer a 9600 bps, lasciando che il DL2000 decida a che velocità comunicare con il modem remoto. Questa prassi garantisce il massimo dell'efficienza del protocollo di compressione a 2400 bps. Infatti, se anche il modem riesce ad arrivare a velocità di 4800 bps grazie alla compressione dati, ma il computer glieli fornisce a 2400, non ci sarà alcuna accelerazione reale. Se invece il computer passa i dati a 9600, l'ampiezza del canale viene sfruttata al meglio. L'handshake modem/computer viene gestito tramite CTS/RTS, vale a dire via hardware. Prima della versione 1.3 delle ROM, il CTS/RTS non era gestito correttamente, vanificando quindi uno dei punti forti del modem. Ora, fortunatamente, tutto è andato a posto. Per finire in bellezza, rendo noto che avremo a breve la release 2.0 della WShell, la shell di Bill Hawes. Sto effettuando il beta-testing per Bill, e sebbene non possa dire per ovvie ragioni cosa c'è dentro, posso garantirvi che sarà una bella sorpresa per tutti. E' invece stata ufficialmente rilasciata da pochi giorni la versione 3.1 di AmigaTeX, che lo riporta alla posizione top tra le versioni disponibili. Infatti, oltre ad implementare TeX 3.0, mette a disposizione l'utilizzo dei font e della grafica PostScript in documenti TeX senza alcun bisogno di avere una stampante dotata di tale linguaggio. L'intera elaborazione PostScript è, infatti, svolta dalla post.library di Adrian Aylward, libreria alla base del celebre interprete PostScript Post. E' possibile includere come immagine un file EPSF (Encapsulated PostScript) in un documento TeX, e mettere su carta il tutto trasparentemente su qualunque stampante. Non ci si poteva del resto aspettare di meno da due dei migliori programmatori su Amiga... ▲

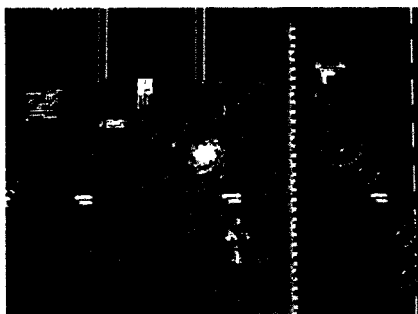
a cura di F. Rossi & S. Crosignani

Spot

Virgin Games

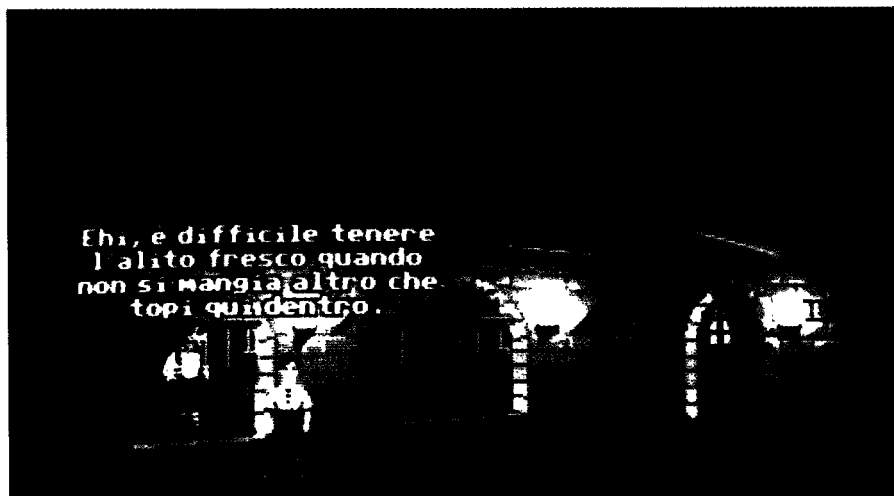
Abbiamo visto giochi tratti da film, da libri, persino da altri giochi... mancava solo un gioco ispirato a una bibita! Spot è il videogioco ufficale del 7-Up, la famosa bibita prodotta dall'onnipotente Coca-Cola Company. Fortunatamente, una volta caricato il dischetto, si scopre (dopo un caricamento piuttosto lungo, a dire il vero) che il programma fortunatamente rifugge da ogni tentativo di pubblicità dozzinale per rivelarsi, invece, un ottimo gioco strategico. Per chi frequenta le sale giochi sarà sufficiente dire che Spot è una versione leggermente migliorata del coin-op Ataxx. Per tutti gli altri... beh, provate a seguire questa spiegazione! Il gioco si svolge su di un tavoliere quadrato dalle caratteristiche variabili: nella configurazione standard è, infatti, possibile utilizzare tutte le caselle, ma è possibile modificare a piacere il piano di gioco inserendovi caselle vuote (e impraticabili) a volontà. In ogni angolo del tavoliere, si trova una pedina colorata; ancora una volta, nella configurazione di base, troviamo due pedine rosse e due blu, ma è possibile far partecipare un massimo di quattro giocatori e quindi di diversi colori. Lo scopo del gioco è di conquistare il maggior numero di caselle praticabili insediandovi pedine del proprio colore. Per far questo, si muovono i propri pezzi iniziali, che quando vengono piazzati in una casella adiacente a un pezzo avversario, lo fanno cambiare di colore (un po' come nell'Othello).
Ci sono due tipi di mosse differenti. Spostando un pezzo in una casella adiacente a quella di partenza si produrrà per meiosi una nuova pedina, che andrà quindi a incrementare il numero di pezzi in gioco. Alternativamente, è possibile muovere fisicamente un pezzo per un massimo di due caselle in ogni direzione, ma in questo caso la posizione iniziale rimarrà pericolosamente sguarnita. Le regole di Spot sono tutte qua, ma come sempre accade nel caso di prodotti dal concetto semplice e intelligente, il gioco risulta estremamente divertente, e senz'altro uno dei migliori titoli ludici che siano stati prodotti per un qualsiasi computer negli ultimi dodici mesi.
Resta tuttavia da chiarire quale sia il legame fra Spot e le bibite: ebbene, quel che non ho detto sinora è che le pedine sono animate! Ogni pezzo, una volta attivato, si trasforma nel pallino rosso con gambe, braccia e occhiali da sole che fa da anni pubblicità al 7-Up, e si sposta da una casella all'altra, sfruttando ogni sorta di mossa di danza. Ci sono moonwalk, break dance, piroette, tuffi e persino passaggi eseguiti su pattini a rotelle o con aste da salto.
A completare questo già bel gioco ci sono, infine, numerose opzioni che permettono di giocare contro avversari computerizzati dal grado di intelligenza differenziata, di configurare i sistemi di comando, di eliminare le animazioni o di generare automaticamente nuove scacchiere. Insomma, c'è tutto quel che si può desiderare da un gioco di strategia. Anche se non vi piace la 7-Up, provate Spot: sono sicuro che non ne rimarrete delusi.

SWIV

Vi ricordate di Silkworm? Sì, era proprio quello stupendo shoot 'em up uscito qualche anno fa, di una tale qualità tecnica, grafica e sonora da essere virtualmente indistinguibile da un gioco da bar.
Ora il team che aveva programmato quella conversione, è tornato alla carica con questo SWIV, che riprende lo schema di gioco del vecchio successo migliorandolo ulteriormente sul piano estetico e di giocabilità.
I protagonisti di questo programma sono due: un elicottero e una jeep capace di saltare e di trasformarsi all'occorrenza in motoscafo.
Entrambi i mezzi vengono visti dall'alto, impegnati in una missione di distruzione pura, durante la quale vengono ostacolati da ogni sorta di arma terrestre, aerea e marina.
Il gioco si svolge su una lunghissima mappa a scorrimento verticale che viene caricata a mano a mano che l'azione procede, dando così luogo a un gioco estremamente vario ma privo di alcuna pausa di attesa. Un'altra caratteristica tecnica estremamente interessante è la "cattiveria" variabile dei nemici, che modificano le loro tattiche di attacco e difesa a seconda dell'abilità dimostrata dal giocatore, permettendo così partite sempre ben equilibrate e divertenti.

Per il resto, c'è ben poco da dire: migliaia di nemici, altrettanti proiettili e gli immancabili bonus per migliorare la propria potenza di fuoco è tutto quel che ci si può aspettare da un gioco di questo tipo. SWIV fornisce i suddetti elementi in quantità, e dovrebbe soddisfare anche il più incontentabile appassionato di questo genere di programmi.

The Secret Of Monkey Island

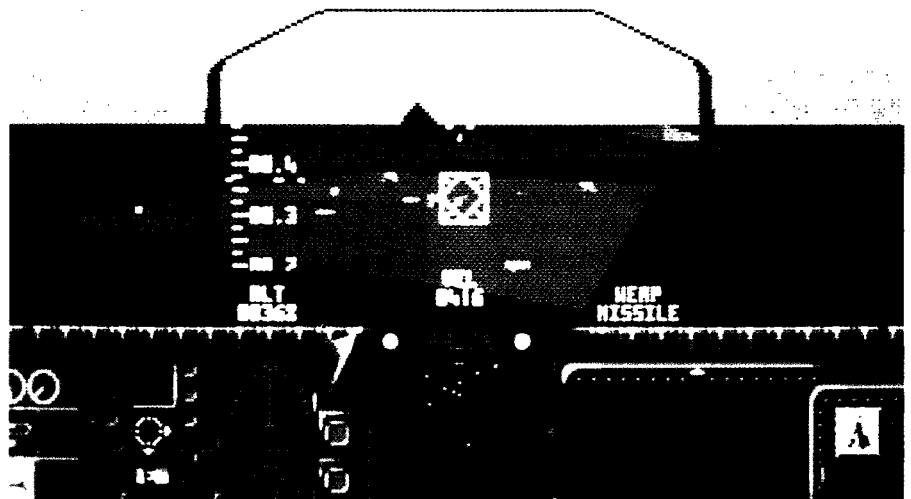
Lucasfilm

Dopo lunghi mesi di assenza dalle scene, la Lucasfilm è finalmente tornata per proporci il suo ultimo capolavoro: un'avventura grafica che segue lo stile dei premiatissimi Zak McKraken e Indiana Jones and The Last Crusade. Il protagonista questa volta è un giovine di belle speranze che deve impegnarsi per superare un esame per poter diventare ufficialmente pirata (dei mari, non dei dischetti). Per fare questo dovrà superare un certo numero di prove ambientate in un arcipelago tropicale, sino a quando non arriverà finalmente a ottenere il comando di un vascello con il quale andare a salvare il governatore delle isole che, una volta tanto, è una stupenda donna. L'ultimo ostacolo che si frappone al matrimonio dei due è "solamente" il fantasma del più crudele pirata mai esistito, che oltretutto è in possesso dell'inquietante Segreto dell'Isola delle Scimmie, un qualcosa dal valore inestimabile. Parlare di questo gioco senza rovinare la sorpresa ai senza dubbio numerosissimi acquirenti, è quasi impossibile: per questo converrà limitarsi a par-

lare della grafica (a cui hanno lavorato decine di artisti) e del sonoro, opera del grandissimo Chris Hulsbeck della Rainbow Arts, forse il miglior musicista Amiga in circolazione. Il sistema di gioco prevede unicamente l'utilizzo del mouse, per cui questa avventura dovrebbe risultare gradita anche ai troppi utenti che ancora non sanno destreggiarsi con la tastiera. Il gioco, che in Italia verrà venduto in versione tradotta, è caratterizzato da un grandissimo umorismo e un'assoluta mancanza di violenza. Sembra incredibile che il mondo dei pirati sia descrivibile anche in questi termini, ma gli autori ci sono riusciti benissimo e giocare con The Secret of Monkey Island assomiglia quasi a seguire un film comico, tanto che a volte si provano azioni negative solo per il gusto di scoprire come possa reagire il proprio personaggio. Per una volta, posso consigliare tranquillamente ai lettori di acquistare il gioco a scatola chiusa. Fatelo, e sono certo che non ve ne pentirete!

Armour-Geddon

Psygnosis

Se gli utenti di Amiga videogiocatori dovessero stilare una classifica delle loro software house preferite, sicuramente la Psygnosis occuperebbe la prima posizione, sia per l'elevata qualità dei suoi prodotti, sia per l'estrema professionalità dimostrata nel lancio dei propri giochi. Era, quindi, inevitabile che queste qualità contraddistinguessero anche l'ultimo capolavoro della software house britannica chiamato, abbastanza apocalitticamente, Armour-geddon.

Cos'è Armour-geddon?
Non è facile dirlo, ma potremmo definirlo un "simulatore vettoriale globale": in poche parole bisogna, al comando di un'armata composta da diversi mezzi, terrestri e non, recuperare cinque pezzi di una bomba a neutroni così da poter distruggere un satellite laser orbitante in grado di spazzare via la Terra in pochi secondi.
I veicoli a propria disposizione sono numerosi, da elicotteri a hovercraft, da carri armati a jet supersonici tutti forniti di un diverso e cospicuo armamento.
Una simulazione così dettagliata e ambiziosa non poteva, ovviamente, fruire di una realizzazione tecnica mediocre e questo, come era lecito aspettarsi dalla Psygnosis, non è accaduto affatto: a partire dalla

superba, fantastica, incerdibile animazione in ray-tracing che accompagna l'introduzione del gioco, si ha l'immediata sensazione di trovarsi di fronte a un futuro classico della storia del videogioco.
Le attese sono comunque premiate una volta che si inizia a giocare: Armour-geddon è realistico, ma immediato, impegnativo, ma divertente, completo e coinvolgente.
La grafica a poligoni è estremamente veloce, il sonoro è ultrarealistico, la giocabilità è incredibile, la longevità pure... Si può volere di più da un videogame?

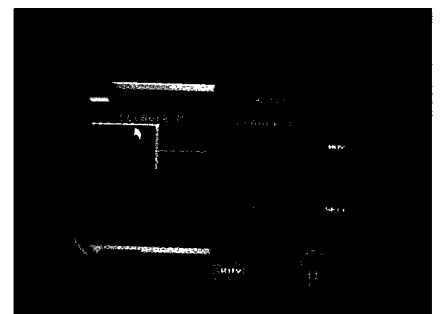
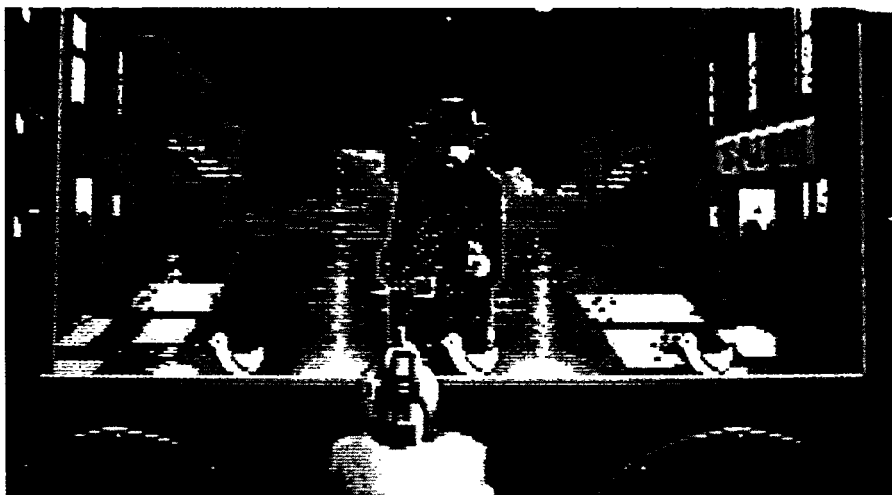
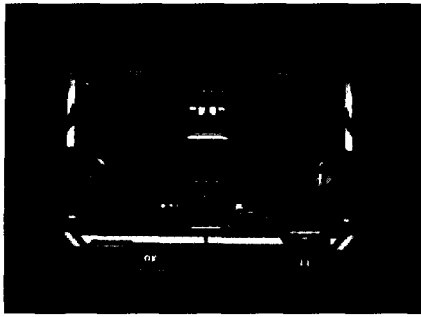
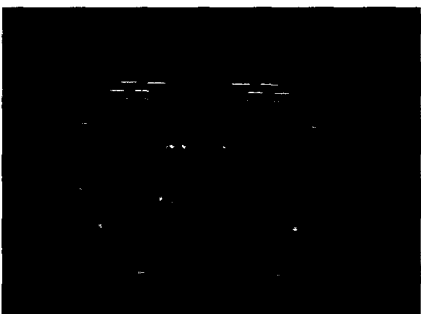
Back to the Future III

Mirrorsoft

Per la serie "avete visto il film e ora giocate al videogioco" ecco a voi Ritorno al futuro 3 o, per i puristi, Back to the future Part III. Non mi dilungherò troppo sulla qualità della pellicola cinematografica (infima, è solo una banale scusa per spillare soldi alla gente...), ma una rinfrescatina per i non amanti del cinema ci vuole: dunque, alla fine della seconda parte avevamo lasciato Doc (l'inventore della DeLorean) nel 1885, mentre Marty (Michael J. Fox) era rimasto solo soletto nel 1955. La situazione non sarebbe poi così malvagia, visto anche che Doc s'è innamorato di Clara Clayton, una maestrina del luogo, se solo non fosse per un piccolo intoppo: o Marty farà un viaggio nel passato per portarsi via lo scienziato pazzo o quest'ultimo finirà con un proiettile nella schiena sparato dal bisbisnonno del nemico numero uno di Marty, Biff. Con una trama così complessa non poteva che scapparci un tie-in multievento e, per la gioia del signor Mirrorsoft, così è stato: quattro sono i livelli con altrettanti compiti e differenti insidie. Il primo stage è "alla Wrath of Demon" con uno scrolling orizzontale da sinistra a destra e Doc che, a cavallo, deve raggiungere la diligenza su cui si trova Clara prima che precipiti nel burrone. Affinché ciò avvenga Doc deve saltare, sparare e schivare tutti gli ostacoli che gli si parano davanti, cowboy e indiani compresi. Il livello successivo è "alla Operation Wolf" con Marty che spara prima alle anatre per allenarsi e poi a delinquenti in carne e ossa. Il livello tre è un piccolo clone del secondo con Marty che tira dei piatti in testa ai cattivoni locali. Infine, l'ultimo stage è alla "nonmiricordopiùilnome" e vede Marty e Doc impegnati sul tetto dei vagoni di un treno, intenti a percorrerlo in tutta la sua lunghezza evitando naturalmente i controllori, gli uomini del malvagio bisbisnonno di Biff, i segnali e i silos per il rifornimento d'acqua.

Metal Masters

Infogrames

Oooh, finalmente sono tornati! Di cosa stiamo parlando?
Ma dei picchiaduro one-o-one, genere di videogames reso famoso da autentici capolavori come Exploding Fist, International Karate, Kung Fu Master & C.
Qualcuno a questo punto si starà immediatamente chiedendo come mai il nome di questo gioco sia Metal Masters se si tratta di una simulazione di arti marziali.
Beh, la spiegazione è quantomai elementare: dovete sì combattere contro un certo numero di nemici, ma dall'interno di un robot, di un mech, di un mobile suit o come diavolo volete chiamarlo.
Ogni macchina è divisa in varie zone (testa, gambe, arti superiori...)

e ognuna di queste ha il proprio indicatore d'energia: quando questo raggiunge un livello critico... Kaboom!
Ovviamente, essendo a bordo di sofisticatissime creature, non ci si limita solo a tirare qualche pugno anzi è possibile usare laser, missilazzi ultrapotenti e fucilate incredibili che, è inutile lo dica, provocano un bel po' di danno fisico al proprio avversario.
In un gioco del genere era inevitabile che ci fosse il solito negozio dove è possibile, sempre che si sia vinto qualche incontro e qualche premio in pecunia sonante, comprare dei pezzi nuovi migliori e sostituire così quelli vecchi e malridotti.
Metal Masters non è niente male: all'inizio l'avevo considerato una semischifezza, ma non appena ho iniziato a giocare l'azione mi ha travolto completamente e mi sono ritrovato megacoinvolto anche a causa degli ultrarealistici "Kachonk!" e "Skatacrash!" che vengono dalle casse dello stereo. Considerando anche l'opzione due giocatori e la presentazione con tanto di parallasse vari ecco che abbiamo tra le mani un prodotto ben al di sopra della sufficienza. Continua così Infogrames!

Supercars 2

Gremlin

Bastaaa! Non ne possiamo più di seguiti, continuazioni, mix, remix e rivincite varie: caspiterina, stiamo parlando di videogiochi non di Twin Peaks! Vabbé visto e considerato che la Gremlin finora non è mai ricorsa a bieche operazioni commerciali del genere li perdoniamo e passiamo a esaminare il loro ultimo prodotto, Supercars... dicevo Supercars 2.
Lo scopo di questo gioco è uno dei più classici, per non dire triti e ritriti, mai apparsi nella storia del videogame: a bordo di un bolide, visto dall'alto, bisogna percorrere un determinato numero di giri della pista e, gareggiando contro le auto avversarie, tagliare per primi il traguardo.
Ovviamente quando si passa all'azione non è tutto così semplice come sembra: prima di tutto le altre auto non se ne stanno a guardare, anzi fanno di tutto per complicarvi la vita urtandovi e facendo così aumentare il livello dei danni del vostro mezzo, livello che, giunto a un certo punto, sancirà l'esplosione dell'auto e la conseguente fine della partita.
Fortunatamente avete a disposizione un bel po' d'armamentario come razzi, missili, ecc. con cui spazzar via la concorrenza. In caso di vittoria riceverete un sacco di soldi con cui potrete cambiare auto, migliorare

quella attuale o comprare optional vari da installare sul vostro mezzo.
Come ho accennato poco fa, l'inquadratura di Supercars è dall'alto con uno scrolling multidirezionale in grado di tenere sempre sotto controllo il vostro bolide. Le principali innovazioni di Supercars 2 rispetto al predecessore sono, manco a farlo apposta, due: la prima è che in caso di partita a due giocatori uno split screen permette a entrambi di correre contemporaneamente e la seconda è la presenza sul tracciato di rampe, salti, ponti e amenità varie.